Troubleshooting Production Planning Problems

- **Troubleshooting:**

  process of identifying the source of problems on the basis of observed trouble symptoms

Troubleshooting Production Planning Problems (Cont.)

- **Problems with production planning can usually be traced to…**
  - **some part of the Planning Subsystem**
  - **problems in other ECS subsystems, including (but not necessarily limited to):**
    - » **Data Processing Subsystem (DPS)**
    - » **Data Server Subsystem (DSS)**
    - » **Communications Subsystem (CSS)**

Troubleshooting Production Planning Problems (Cont.)

- **Fault Recovery**
  - **Each request that crosses a client/server boundary is assigned a system-unique identifier referred to as an RPC ID**
  - **The RPC ID facilitates the automatic fault recovery events that occur whenever there is a client or server failure**
  - **As a request propagates through the system, each associated client/server exchange is assigned a unique RPC ID**
    - » **The RPC ID for each interaction is derived from the previous RPC ID received by the client for the request; consequently, all RPC IDs associated with a given request have a common portion that relates the various client/server calls to one another**
    - » **Given the previous RPC ID, clients consistently reproduce the same RPC ID that was submitted to the server on the subsequent event**

Troubleshooting Production Planning Problems (Cont.)

- **Fault Recovery (Cont.)**
  - The concept of reproducible RPC IDs is central to the ECS fault recovery capability
    - When requests are retried from client to server, they are always submitted with the same RPC ID that was used in the original submission of the request, even if either client or server has crashed between retries
  - The RPC ID is also central to the check-pointing aspect of fault recovery
    - As requests arrive at fault recovery-enabled servers, they are recorded in a persistent store (typically a database), tagged with the RPC ID
    - As the request is serviced, check-pointing state information may be updated in the persistent store, up to and including the request's completion status
    - This allows the servers to resume servicing from the last check-pointed state, particularly upon resubmission from a client

Troubleshooting Production Planning Problems (Cont.)

- **Fault Recovery (Cont.)**
  - **PLANG and PRONG components check-point the following types of information:**
    - » **EcPlSubMgr - Unprocessed subscription notifications**
    - » **EcDpPrDeletion - Interim Delete Requests**
    - » **EcDpPrEM - Queued and activated jobs**

Troubleshooting Production Planning Problems (Cont.)

- **Fault Recovery: Fault Handling**
  - **Failure events are classified according to the following three severity levels:**
    - » **Fatal error**
    - » **Retry error**
    - » **Warning**

Troubleshooting Production Planning Problems (Cont.)

- **Fault Recovery: Fault Handling (Cont.)**
  - **Fatal error is returned when a request cannot be serviced, even with operator intervention**
    - » **For example, if a request is made to distribute data via ftp to a non-existent host, the request is failed**
  - **Retry error is a potentially recoverable error**
    - » **Normally, a retry error would be returned to the client only when the server cannot recover from the error automatically**
    - » **A retry error may require operator assistance**
    - » **For example, entering a new name for a PR after being notified that a previously entered name contained too many characters**
  - **Warning is provided when operations can proceed but an unexpected circumstance was detected**
    - » **For example, if a client requests removal of a file but the file does not exist**

Troubleshooting Production Planning Problems (Cont.)

- **Fault Recovery: Fault Handling (Cont.)**
  - **Transient errors (such as network errors) are always retry errors**
    - » **In general, clients and servers that experience transient retry errors first attempt to recover by retrying the operation automatically**
    - » **One special case of this is "rebinding," which refers to the process by which a client automatically attempts to re-establish communication with a server in the event communication is disrupted**
    - » **The disruption may be caused by transient network failure, or by the server crashing or being brought down**
    - » **In any case, the client automatically attempts to reconnect to the server for a configurable period of time on a client-by-client basis**

Troubleshooting Production Planning Problems (Cont.)

- **Fault Recovery: Fault Handling (Cont.)**
  - **ECS processes encountering an error or receiving an error from a server request can either pass the error back to a higher-level client or present it to the operator for operator intervention**
  - **The specific fault handling policies for PLANG and PRONG client processes are shown in the table that follows**

Troubleshooting Production Planning Problems (Cont.)

PLANG and PRONG Fault Handling Policies

Client Process	Fault Handling Policy
EcPlSubMgr	**Retry errors:** All Subscription processing errors are retried a configurable number of times and for a configurable time period. After the configurable number of times (or time period) the subscription is lost. **Fatal errors:** N/A.
EcPlPREditor_IF **EcPlWb**	**Retry errors:** Since these are GUI applications, errors are reported to the operator and it is the operator's responsibility to retry the request. **Fatal errors:** Errors are reported to the operator.
EcPlOdMgr	**Retry errors:** Retries errors from the Science Data Server and the Subscription Server. **Fatal errors:** Logs errors and stops current on demand requests.
EcDpPrEM	**Retry errors:** Errors are retried a configurable number of times, then the job is failed and it is up to the Production Monitor to restart the job through AutoSys. **Fatal errors:** A fatal error message is logged.
EcDpPrJobMgmt	**Retry errors:** If a DPR cannot be assigned to a machine or created in AutoSys, it is left in a PENDING state and the assignment is retried after DpPrPendingThreadWaitInterval seconds. **Fatal errors:** N/A.

Troubleshooting Production Planning Problems (Cont.)

PLANG and PRONG Fault Handling Policies (Cont.)

Client Process	Fault Handling Policy
EcDpPrDeletion	**Retry errors:** No retries are implemented. Status from DSS is not checked. **Fatal errors:** N/A.

Troubleshooting Production Planning Problems (Cont.)

- **Fault Recovery: Client Crash and Restart**
  - **When a client of a PLANG or PRONG server crashes, the server (i.e., EcPlSubMgr, EcDpPrJobMgmt, or EcDpPrDeletion) continues to service the requests that were in process at the time of the client's crash**
  - **When a client restarts in the ECS system, it sends a restart notification to each server with which it interacts**
    - » **Clients notify servers that they have come up either "cold" or "warm"**
    - » **Generally, the notification temperature sent to the server matches the temperature at which the client process is restarted**

Troubleshooting Production Planning Problems (Cont.)

- **Fault Recovery: Client Crash and Restart (Cont.)**
  - **Default server behavior in response to "warm" startup notification from a client:**
    - » **Outstanding requests for the restarted clients remain available in the persistent store**
    - » **The outstanding requests may be resubmitted by the client, and are serviced to completion upon resubmission**
    - » **Associated resources are left allocated until the requests are completed**

Troubleshooting Production Planning Problems (Cont.)

- **Fault Recovery: Client Crash and Restart (Cont.)**
  - **Default server behavior in response to "cold" startup notification from a client:**
    - » **All outstanding requests for the restarted client are cancelled**
    - » **If the client resubmits any cancelled request using the same RPC ID (e.g., by pressing the Retry button from an operator GUI), it is failed with a fatal error due to the client cold startup notification**
    - » **Any resources associated with the cancelled requests are released and reclaimed by the system**

Troubleshooting Production Planning Problems (Cont.)

- **Fault Recovery: Server Crash and Restart**
  - **When a server crashes, clients cannot continue to submit requests for processing**
  - **Synchronous requests in progress result in a Distributed Computing Environment (DCE) exception being thrown back to the client process, which enters a rebinding failure recovery mode (as previously mentioned)**
  - **Attempts to submit requests while the server is down result in the client blocking until a communication timeout has been reached**
  - **Although DCE has been replaced by socket-based library calls (i.e., CCS Middleware), the DCE exception code is handled by the CCS Middleware**

Troubleshooting Production Planning Problems (Cont.)

- **Fault Recovery: Server Crash and Restart (Cont.)**
  - When a server restarts, it may perform various resynchronization activities in order to recover from an unexpected termination
  - In the event of a server cold start or cold restart, the server typically cancels all outstanding requests and reclaims all associated resources
  - In general, existing request queues are retained for warm restarts and cleared for cold starts or cold restarts

Troubleshooting Production Planning Problems (Cont.)

- **Fault Recovery: Server Crash and Restart (Cont.)**
  - **EcPlSubMgr-specific activities upon start/restart:**
    - » **Warm Restart: Any subscriptions that have not been processed are read from checkpoint file and processed**
    - » **Cold Start or Cold Restart: N/A**
  - **EcDpPrJobMgmt-specific activities upon start/restart:**
    - » **Warm Restart: Jobs in AutoSys and jobs waiting in the queue are read from the database; any jobs that are ready are placed into AutoSys from the queue (if there are processing slots available)**
    - » **Cold Start or Cold Restart: N/A**

Troubleshooting Production Planning Problems (Cont.)

- **Fault Recovery: Server Crash and Restart (Cont.)**
  - **EcDpPrDeletion-specific activities upon start/restart:**
    - » **Warm Restart: Interim granules marked for deletion are read from the database and are deleted when time-out occurs**
    - » **Cold Start or Cold Restart: N/A**

Troubleshooting Production Planning Problems (Cont.)

- **Fault Recovery: Request Resubmission**
  - **Upon restarting a crashed client or server, requests are typically resubmitted**
  - **If the restarted process was started warm, the fault-recovery capabilities permit the server to resume processing of the request from its last check-pointed state**
    - » **This prevents needless repetition of potentially time-consuming activities**
  - **EcDpPrJobMgmt- and EcDpPrDeletion-specific activities upon resubmission of a request:**
    - » **Requests are submitted synchronously**
    - » **If the entire request is resubmitted by a client, then only that part of the resubmitted request that has not been completed is reprocessed**

Troubleshooting Production Planning Problems (Cont.)

- **Troubleshooting table**
  - **describes actions to be taken in response to some common Production Planning problems**
  - **if the problem cannot be identified and fixed without help within a reasonable period of time, call the help desk and submit a trouble ticket in accordance with site Problem Management policy**

Troubleshooting Production Planning Problems (Cont.)

Symptom	Response
Unable to log in to the Planning Subsystem host (e.g., g0pls01).	Check with the Operations Controller/System Administrator to ensure that the host is “up."
GUI not displayed when the start-up script has been properly invoked.	Ensure that the DISPLAY variable was set properly. [For detailed instructions refer to the applicable procedure, either **Launching the Production Request Editor** or **Launching Planning Workbench-Related GUIs** (previous sections of this lesson).]
Error message indicating that SNS (System Name Server) and/or Resource Model is/are in use using the selected Application ID.	1. Use another Application ID if working in a different mode from the person using the selected Application ID. 2. If working in the same mode as the other user, coordinate use of Planning applications with the other user and/or the System Administrator. [For detailed instructions refer to the procedure for **Launching Planning Workbench-Related GUIs** (previous section of this lesson).]
Error message associated with the Production Request Editor.	Refer to Table 3, Production Request Editor User Messages (adapted from the corresponding table in 609-CD-610-003, *Release 6B Operations Tools Manual for the ECS Project*).
Error message associated with the Production Strategies GUI.	Refer to Table 4, Production Strategy User Messages (adapted from the corresponding table in 609-CD-610-003, *Release 6B Operations Tools Manual for the ECS Project*).

Troubleshooting Production Planning Problems (Cont.)

Symptom	Response
Error message associated with the Planning Workbench.	Refer to Table 5, Planning Workbench User Messages (adapted from the corresponding table in 609-CD-610-003, *Release 6B Operations Tools Manual for the ECS Project*).
Production Request fails (DPR generation fails).	1. Ensure that it is possible to connect to the necessary hosts and servers (listed in Table 6). [For detailed instructions refer to the section on **Checking Connections to Hosts/Servers** (subsequent section of this lesson).] 2. If hosts/servers are all "up," perform the procedure for **Handling a Failure to Generate DPRs** (subsequent section of this lesson). 3. Retry generating DPRs by resaving the Production Request. [For detailed instructions refer to the section on **Editing/Modifying a Production Request** (previous section of this lesson).]

Troubleshooting Production Planning Problems (Cont.)

Symptom	Response
PR or DPR deletion hangs.	1. Ensure that enough time has passed to allow DPR deletion (deleting a DPR can require as much time as creating a DPR). 2. Ensure that it is possible to connect to the necessary hosts and servers (listed in Table 6). (Both the Job Management Server and Deletion Server are called to clean up all PDPS database tables associated with the DPR or PR.) [For detailed instructions refer to the section on **Checking Connections to Hosts/Servers** (subsequent section of this lesson).] 3. If hosts/servers are all "up," check for a database lock or resource lock in the PDPS database. [For detailed instructions refer to the section on **Responding to PR or DPR Deletion that Hangs** (subsequent section of this lesson).]

Troubleshooting Production Planning Problems (Cont.)

Symptom	Response
DPR deletion fails.	1. Ensure that enough time has passed to allow DPR deletion (deleting a DPR can require as much time as creating a DPR). 2. Ensure that it is possible to connect to the necessary hosts and servers (listed in Table 6). (Both the Job Management Server and Deletion Server are called to clean up all PDPS database tables associated with the DPR or PR.) [For detailed instructions refer to the section on **Checking Connections to Hosts/Servers** (subsequent section of this lesson).] 3. If hosts/servers are all "up," check the Deletion Server Debug log (EcDpPrDeletionDebug.log). [For detailed instructions refer to the section on **Responding to DPR Deletion that Fails** (subsequent section of this lesson).]

Troubleshooting Production Planning Problems (Cont.)

Symptom	Response
DPR scheduling fails (DPR is not passed to Data Processing).	1. Ensure that it is possible to connect to the necessary hosts and servers (listed in Table 6). [For detailed instructions refer to the section on **Checking Connections to Hosts/Servers** (subsequent section of this lesson).] 2. If hosts/servers are all "up," perform the procedure for **Handling a DPR Scheduling Failure** (subsequent section of this lesson). 3. If necessary, delete the affected DPRs. [For detailed instructions refer to the section on **Creating a New Production Plan** (previous section of this lesson).] 4. If affected DPRs were deleted, recreate the DPRs. [For detailed instructions refer to the section on **Editing/Modifying a Production Request** (previous section of this lesson).] 5. If affected DPRs were recreated, create a new production plan. [For detailed instructions refer to the section on **Creating a New Production Plan** (previous section of this lesson).]

Troubleshooting Production Planning Problems (Cont.)

Symptom	Response
Other problems.	1. Ensure that it is possible to connect to the necessary hosts and servers (listed in Table 6). [For detailed instructions refer to the section on **Checking Connections to Hosts/Servers** (subsequent section of this lesson).] 2. If hosts/servers are all "up," check the log files (e.g., EcPlPREditor.ALOG, EcPlPREditorDebug.log, EcPlWb.ALOG, EcPlWbDebug.log, EcPlTI.ALOG) in the /usr/ecs/*MODE*/CUSTOM/logs directory for error messages. [For detailed instructions refer to the procedure for **Checking Log Files** (subsequent section of this lesson).]

Hosts, Servers, etc. Relevant to Planning & Processing

HOST	SERVER/CLIENT/OTHER SOFTWARE
Planning/Management Workstation	Production Request Editor (EcPlPREditor) Planning Workbench GUI (EcPlWb) Production Strategies GUI (EcPlProdStrat) Production Planning Master Timeline (EcPlTl) Message Handler (EcPlMsh) System Name Server (EcPlSns) Resource Model (EcPlRm)
Queuing Server (e.g., x0sps02)	Job Management Server (EcDpPrJobMgmt) Deletion Server (EcDpPrDeletion) Execution Management (EcDpPrEM) AutoSys Event Processor (event_demon) AutoSys Event Server (Sybase server) (e.g., x0sps02_srvr) On-Demand Manager (EcPlOdMgr) Subscription Manager (EcPlSubMgr) PDPS database Sybase server (e.g., x0sps02_srvr)
Science Processor (e.g., x0spg01)	PGE Management (EcDpPrRunPGE) Resource Usage (EcDpPrRusage) PGE

Hosts, Servers, etc. Relevant to Planning & Processing (Cont.)

HOST	SERVER/CLIENT/OTHER SOFTWARE
Access/Process Coordinators (APC) Server (e.g., x0acg01)	Archive Server (EcDsStArchiveServer) FTP Server (EcDsStFtpServer) Cache Manager Server (EcDsStCacheManagerServer) Staging Disk Server (EcDsStStagingDiskServer) Pull Monitor Server (EcDsStPullMonitorServer)
Ingest Server (e.g., x0icg01)	Name Server (EcCsIdNameServer) Registry Server (EcCsRegistry)
Sun internal server (e.g., x0acs06)	Science Data Server (EcDsScienceDataServer) Data Dictionary (EcDmDictServer) Subscription Server (EcSbSubServer) Event Server (EcSbEventServer)

Troubleshooting Production Planning Problems (Cont.)

- **Production Planning Troubleshooting Procedures**
  - Checking Connections to Hosts/Servers
  - Checking the Production Request Editor ALOG File
  - Using ISQL to Check Database Tables
  - Checking the PDPS Database for Causes of Failure to Generate DPRs
  - Checking the Production Request Editor Debug File for Evidence of Metadata Queries
  - Checking for Database Deadlocks
  - Checking for Resource Locks in the PDPS Database

Troubleshooting Production Planning Problems (Cont.)

- **Production Planning Troubleshooting Procedures (Cont.)**
  - **Responding to DPR Deletion that Fails**
  - **Responding to a "DPR Validation Failed" Error**
  - **Responding to an "information (INFO) Production Request {Production Request Id} has unschedulable DPR {DPR Id}" Error**
  - **Checking Log Files**
  - **Checking Database Connections**

Troubleshooting Production Planning Problems (Cont.)

- **Procedure (Checking Connections to Hosts/Servers):**
  - **Access a terminal window logged in to the Planning/Management Workstation host**
  - **Change directory to the utilities subdirectory (/usr/ecs/MODE/CUSTOM/utilities)**
  - **At the command line prompt enter EcCsIdPingServers *MODE***
  - **Observe the results displayed on the screen to determine whether connections can be made with the necessary hosts and servers**
  - **Ping the servers again (EcCsIdPingServers *MODE*)**
  - **If it is not possible to connect to any needed host(s)/server(s), notify the Operations Controller/System Administrator to check the hosts/servers and bring them back up if necessary**

Troubleshooting Production Planning Problems (Cont.)

- **Handling a Failure to Generate DPRs**
  - Checking the Production Request Editor ALOG File
  - Using ISQL to check database tables
  - Checking the PDPS database for causes of failure to generate DPRs
  - Determining whether DPR explosion fails because Production Request Editor does not query DSS for data
  - Checking the Production Request Editor debug file for evidence of metadata queries
- **Responding to PR or DPR Deletion that Hangs**
  - Checking for database deadlocks
  - Checking for resource locks in the PDPS database

Troubleshooting Production Planning Problems (Cont.)

- Responding to DPR Deletion that Fails
- Handling a DPR Scheduling failure
  - Responding to a "DPR Validation Failed" error
  - Responding to an "information (INFO) Production Request {Production Request Id} has unschedulable DPR {DPR Id}" error
- Checking Log Files
- Checking Database Connections

Production Processing

- **DPR Chaining**
  - **The Job Management Server in the DPS schedules chains of DPRs through the use of stored procedures in the PDPS database**
  - **In addition, when managing chains of DPRs, Job Management uses data from the following two tables in the PDPS database:**
    - » **DpPrPgeLimits**
    - » **DpPrClassSchedulingLimits**

Production Processing (Cont.)

- DPR Chaining (Cont.)
  - DpPrPgeLimits imposes restrictions on the number of chain-head DPRs of a particular PGE that can run simultaneously on the same virtual machine
    - » A database record defines each PGE-virtual computer combination that will be run
  - DpPrPgeLimits table has the following columns:
    - » pgeId - PGE ID
    - » computerName – virtual computer (vc) name
    - » maxConcurrent - defines the maximum number of chain heads with a particular PGE ID that can run at the same time on a specific vc
    - » numConcurrent - shows how many chain heads with a particular PGE ID are running at the same time on a specific vc
    - » numScheduled - shows how many chain heads with a particular PGE ID are currently scheduled on a vc

Production Processing (Cont.)

- DPR Chaining (Cont.)
  - If there were no record in the DpPrPgeLimits table for a particular PGE-computer combination that was scheduled, there would be no limits on the DPRs for that PGE
    - » PGE would be able to run and there would be no limits placed on how many DPRs for that PGE could run on the same virtual machine
    - » DPRs would run on the machine specified by the Production Planner in the Production Request (if the Production Planner designated a machine when creating the Production Request)
    - » If no machine was specified in the Production Request, the machine would be determined from the PlResourceRequirement table (data entered during PGE registration)

Production Processing (Cont.)

- DPR Chaining (Cont.)
  - DpPrClassSchedulingLimits limits the classes of DPRs that can run at any point in time
    - » Classes correspond to the types of processing
  - DpPrClassSchedulingLimits table has three records, one for each type of processing
  - Each record has the following fields:
    - » dprClass - assigned value identifies the type of processing (0 = Routine Processing, 1 = On-Demand Processing, 2 = Reprocessing)
    - » maxDprs - maximum number of jobs of the type (in dprClass) that are allowed to run on the system
    - » minDprs - currently not used
    - » currentDprs - number of jobs of the type (in dprClass) that are currently running

Production Processing (Cont.)

- **DPR Chaining (Cont.)**
  - **If the DpPrClassSchedulingLimits table has no record for a particular type of processing, DPRs of that type are not allowed into AutoSys**
  - **Values for the maxDprs and minDprs columns in the DpPrClassSchedulingLimits table are loaded at Job Management Server startup using data from two configuration parameters:**
    - » **DpPrMaxConcurrentDPRs - maximum allowed jobs**
    - » **DpPrMinConcurrentDPRs - minimum allowed jobs**
  - **Each parameter has three integer values; the first for routine processing; the second for on-demand processing; and the third for reprocessing jobs**

Production Processing (Cont.)

- DPR Chaining (Cont.)
  - Example: Configuration Registry may have the following entries:
    - » DpPrMaxConcurrentDPRs = 100 60 40
    - » DpPrMinConcurrentDPRs = 0 0 0
  - In this case the maximum allowed jobs is 100 for routine processing, 60 for on-demand processing, and 40 for reprocessing
  - Minimum allowed jobs is 0 for each type of processing

Production Processing (Cont.)

- DPR Chaining (Cont.)
  - Total number of completed jobs allowed in AutoSys is defined by a configuration parameter:
    - » DpPrAutoSysMaxDPRs
  - Within the restrictions of the DpPrClassSchedulingLimits and DpPrPgeLimits database tables classes of DPR chains are scheduled by DPS with chains of DPRs (identified by their chain heads) being scheduled on the same machine whenever possible
  - When scheduling a chain-head DPR, attempts are made to schedule it on the machine that has the highest number of accepted inputs if accepted inputs are found
  - The chain-head DPR's children should be scheduled on the same machine

Production Processing (Cont.)

- DPR Chaining (Cont.)
  - Load balancing is done among virtual computers on the same string when selecting a computer for a DPR chain
  - If a DPR does not have its PGE identified in the DpPrPgeLimits table, that does not prevent it from being scheduled on a particular machine if most of the DPR's inputs are staged on that machine
  - If there is more than one chain head for a set of DPRs, the DPRs are combined into a single chain
  - For example:
    - » DPR#1 and DPR#2 are both parents of DPR#3
    - » Marking DPRs#1 and #2 as chain heads results in a single chain consisting of all three DPRs

Production Processing (Cont.)

- DPR Chaining (Cont.)
  - Another example (MODIS processing:
    - » If the Production Planner defines both the MODPGE01 PR and the MODPGE02 PR as chain heads, they are combined in one chain with the MODPGE01 DPR ID as the chainId
    - » MODPGE01 is designated the chain head because its outputs are used as inputs to MODPGE02 and MODPGE08
    - » MODPGE02 outputs are used as input to MODPGE08
  - The PGEs that will be used as chain heads should be identified before installation
  - In selecting potential chain heads, it is recommended that PGEs which create substantial (i.e., large or many) inputs for other PGEs be so designated

Production Processing (Cont.)

- **Copy on Demand Feature**
  - **A feature whereby the DPS code stages granules locally; i.e., granule files are staged to the processor on which the PGE is running**
  - **The copy on demand feature is specified in the following locations:**
    - » **onDemandCopy column in the PlDataTypeMaster database table (PDPS database) (the value of the onDemandCopy flag is used by the system to determine whether input files need to be locally staged)**
    - » **ON_DEMAND_COPY flag in the ESDT ODL files after DATA_TYPE and DATA_TYPE_VERSION; for example:**

      ON_DEMAND_COPY = "Y"

Production Processing (Cont.)

- **DPR Output Files Immediately Available as Input**
  - **Output files are immediately available as input when generated as output of a DPR**
  - **There is no waiting for Subscription Server notification before the output of a parent DPR is used as input to a child DPR**

Production Processing (Cont.)

- **AutoSys/AutoXpert**
  - **production scheduling tool**
  - **supports the operational activities surrounding production processing in the PDPS**
  - **assists with the following activities (among others):**
    - » **job monitoring**
    - » **job scheduling**
    - » **fault notification**
    - » **job restart**
    - » **determining the effects of failure of a DPR**
    - » **determining the cause and actions to be taken due to the failure of a DPR**

Production Processing (Cont.)

- **AutoSys/AutoXpert (Cont.)**
  - **displays DPRs as job boxes**
  - **recognizes the following three categories of jobs:**
    - » **box jobs**
    - » **command jobs**
    - » **file-watcher jobs**

Production Processing (Cont.)

- **Box job**
  - collection of other jobs
  - provides an organizational structure for a group of jobs that should be run within the same time period
    - » performs no processing action
- **Command job**
  - "command" can be a shell script, the name of an executable program, a file transfer, or any other command that causes execution of a UNIX command on client machine

Production Processing (Cont.)

- **File-watcher job**
  - **functions similarly to a command job**
  - **monitors the creation and size of a particular operating system file**
  - **allows AutoSys to know the status of external files that are needed in the processing of command jobs or box jobs**

Production Processing (Cont.)

- **AutoSys Job Starting Parameters**
  - **Date and time scheduling parameters are met**
  - **Starting Conditions specified in the job definition evaluate to "true"**
  - **For jobs in a box, the box must be in the RUNNING state**
  - **The current status of the job is not ON_HOLD or ON_ICE**
- **AutoSys finds all jobs that may be affected by any change in the truth of the starting parameters and determines whether or not to start the jobs**

Production Processing (Cont.)

- AutoSys Jobs (ECS)
  - Each DPR generated by the Planning Subsystem defines a box job for AutoSys
  - Every DPR/box job is composed of three command jobs that run in the following order:
    - » Preprocessing (EcDpPrEM)
    - » Execution (EcDpPrRunPGE)
    - » Postprocessing (EcDpPrEM)
  - The number of command jobs in an AutoSys Job box was reduced from seven to three in order to reduce overhead
    - » Reduces the load on the AutoSys Event Processor by about one half
    - » Has no effect on the DPS Queuing Server CPU loading

Production Processing (Cont.)

- AutoSys Jobs (ECS)
  - To create the new Preprocessing job the following old command jobs were combined:
    - » Allocation
    - » Staging
    - » Pre-processing
  - To create the new Postprocessing job the following old command jobs were combined:
    - » Post-processing
    - » Destaging
    - » Deallocation
  - The value in the database column dprState correlates to the seven old job states

Production Processing (Cont.)

- **ECS Job Names**
  - **Indicate the PGE and the mode in which the DPR was generated and the stage of processing**
  - **Example:**
    - » **AM1Eph#2.012302200TS2**
    - » **AM1Eph#2.012302200TS2R**
    - » **AM1Eph#2.012302200TS2E**
    - » **AM1Eph#2.012302200TS2P**

Production Processing (Cont.)

- **ECS Job Names (Cont.)**
  - **The first job name in the list is a DPR/box job-level name**
    - » **First few characters identify the PGE (i.e., AM1Eph - Step 1 DPREP)**
    - » **Last three characters of the DPR/box job-level name (i.e., TS2) indicate the mode in which the DPR was generated**
    - » **Last four characters of the remaining (command) job names in the list indicate the mode (i.e., TS2) and the stage of processing**
    - » **Job name ending in "R" is the preprocessing job; job name ending in "E" is the execution job; job name ending in "P" is the postprocessing job**

Production Processing (Cont.)

- Preprocessing depends on the box job having started
- Execution depends on successful completion of the command job that precedes it (preprocessing)
- Postprocessing depends on Execution having completed (not necessarily “successful,” just done)
  - If execution failed, postprocessing handles failed PGE processing

Production Processing (Cont.)

- DPR (box job) may be dependent on the successful completion of some other DPR
  - usually involving a need for the output of another DPR as input
- Effects of DPR dependencies
  - dependent DPRs are "held" by AutoSys until their data availability subscriptions are fulfilled
  - subscription manager software (in the PLS) informs the DPS to release the AutoSys jobs after all data subscriptions for a given DPR are fulfilled
  - DPS (as monitored by the AutoSys Job Scheduling engine) runs the PGEs and associated jobs as the resources required for the tasks become available
  - procedure continues until all DPRs scheduled for the day have completed

Production Processing (Cont.)

- **Each mode has multiple Job Management queues**
  - **One queue for each type of processing**
  - **There is a maximum number of jobs allowed in AutoSys for each type of processing**
  - **For example, in each mode a DAAC would have queues for the following types of processing:**
    - » **Routine Processing**
    - » **On-Demand Processing**
    - » **Reprocessing**

Production Processing (Cont.)

- **DAAC Production Monitor uses AutoSys/ AutoXpert when monitoring and controlling job processing, including the following functions:**
  - **Deleting/suspending/resuming jobs as required**
  - **Monitoring and providing processing status upon request**

Launching the AutoSys GUI Control Panel

- **Production Processing Applications**
  - **Subscription Manager**
  - **Job Management**
  - **Execution Management**
  - **PGE Management**
  - **Deletion Server**
  - **AutoSys/AutoXpert**
    - » **Event Processor**
    - » **Event Server**
    - » **AutoSys GUIs**
    - » **AutoXpert GUIs**

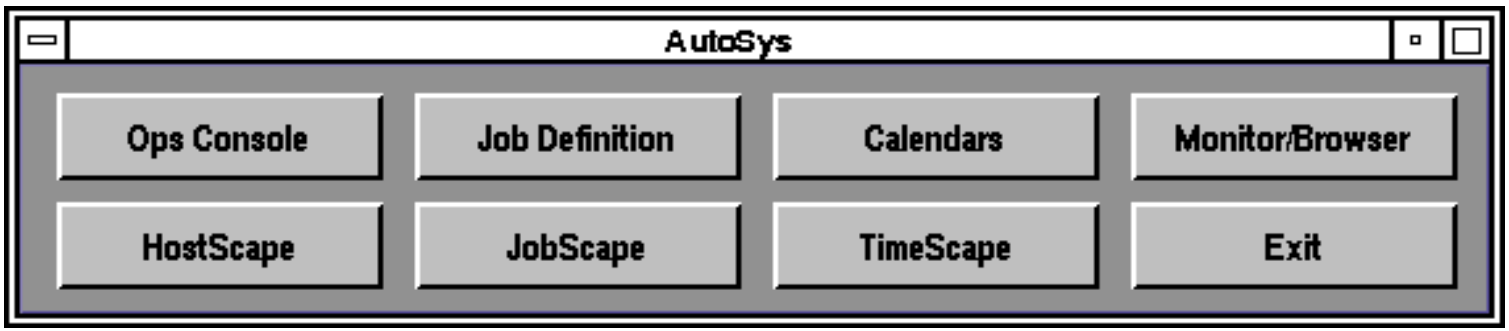
Launching the AutoSys GUI Control Panel

- **Production Processing Applications (Cont.)**
  - **QA Monitor**
  - **Sybase ASE Server**

Launching the AutoSys GUI Control Panel (Cont.)

- **Procedure**
  - Access a terminal window logged in to the Queuing Server host
  - Set the ECS_HOME environmental variable if necessary
  - Change directory to the subdirectory (e.g., autouser) containing the set-up files
  - Source the appropriate file
  - Change directory to the subdirectory (e.g., utilities) containing the AutoSys start script
  - Start AutoSys in the appropriate mode

AutoSys GUI Control Panel



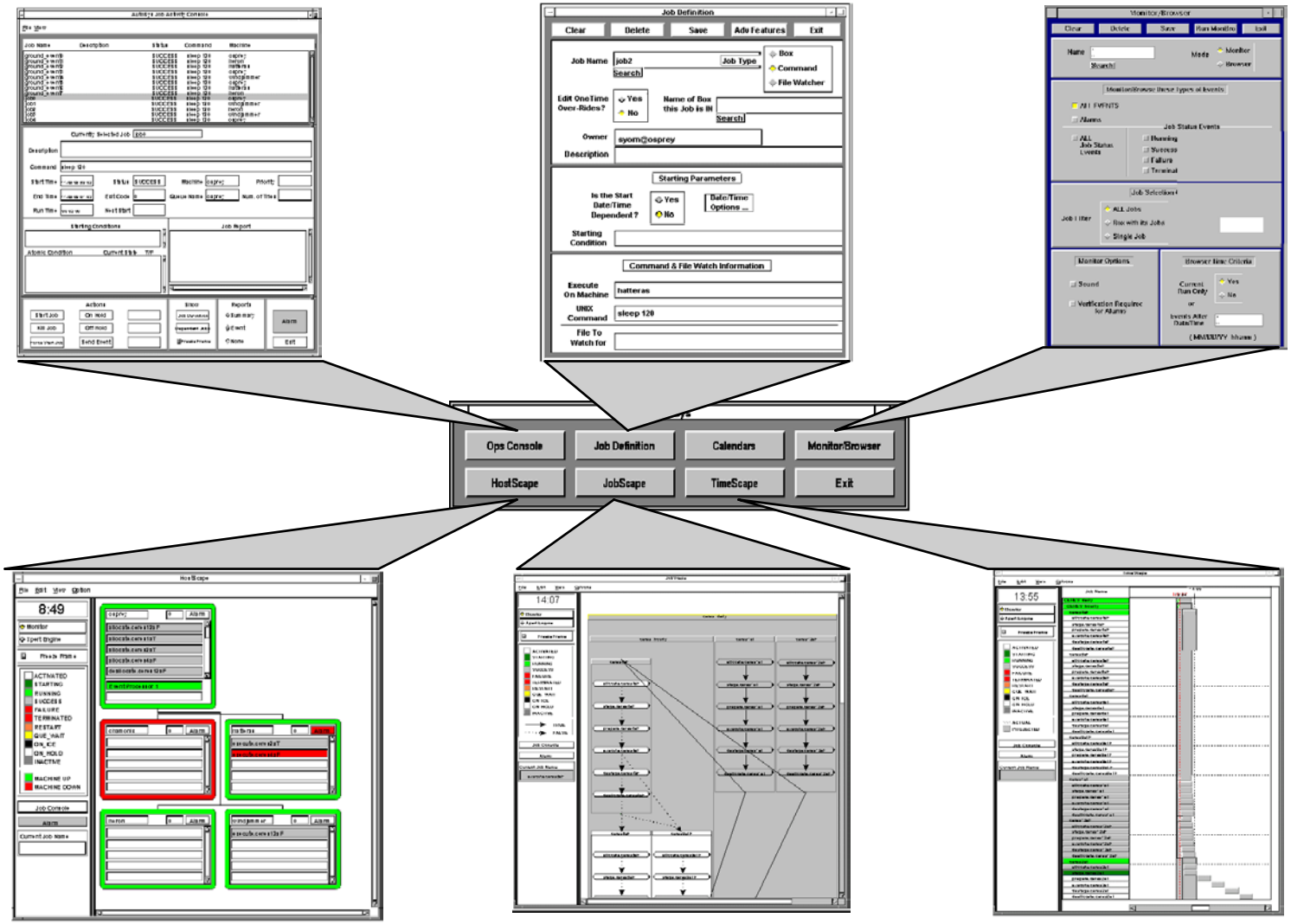
Configuring AutoSys Screens/Displays

- **Configuring AutoSys Runtime Options**
  - **Refresh Interval**
    - » **determines how often the View Region will be updated**
  - **Ping Interval**
    - » **defines how often the connectivity will be evaluated**
  - **Hang Time**
    - » **specifies the length of time jobs will continue to be displayed within a machine after they have completed running**
  - **Inches/Hr**
    - » **indicates how much information is displayed on the screen**

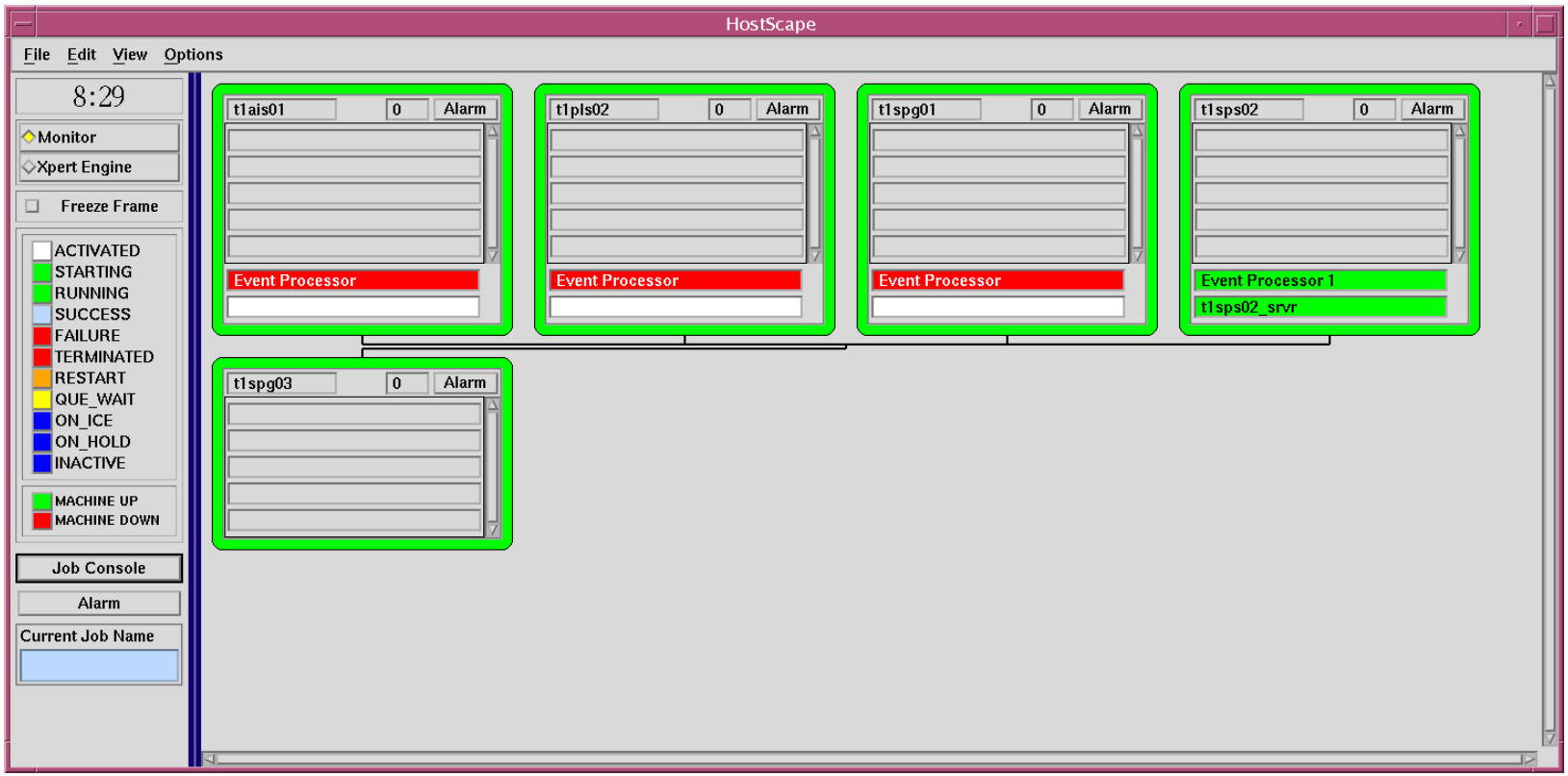
Configuring AutoSys (Cont.): Configuring Runtime Options

- **Procedure**
  - Click on either the HostScape, TimeScape, or JobScape button to display the corresponding AutoXpert GUI
  - Select Options → Edit Runtime Options from the pull-down menu to display the Runtime Options dialog box
  - Enter new values for the runtime options as necessary
    - » refresh interval
    - » ping interval
    - » hang time
    - » number of inches/hour
  - Apply the modifications

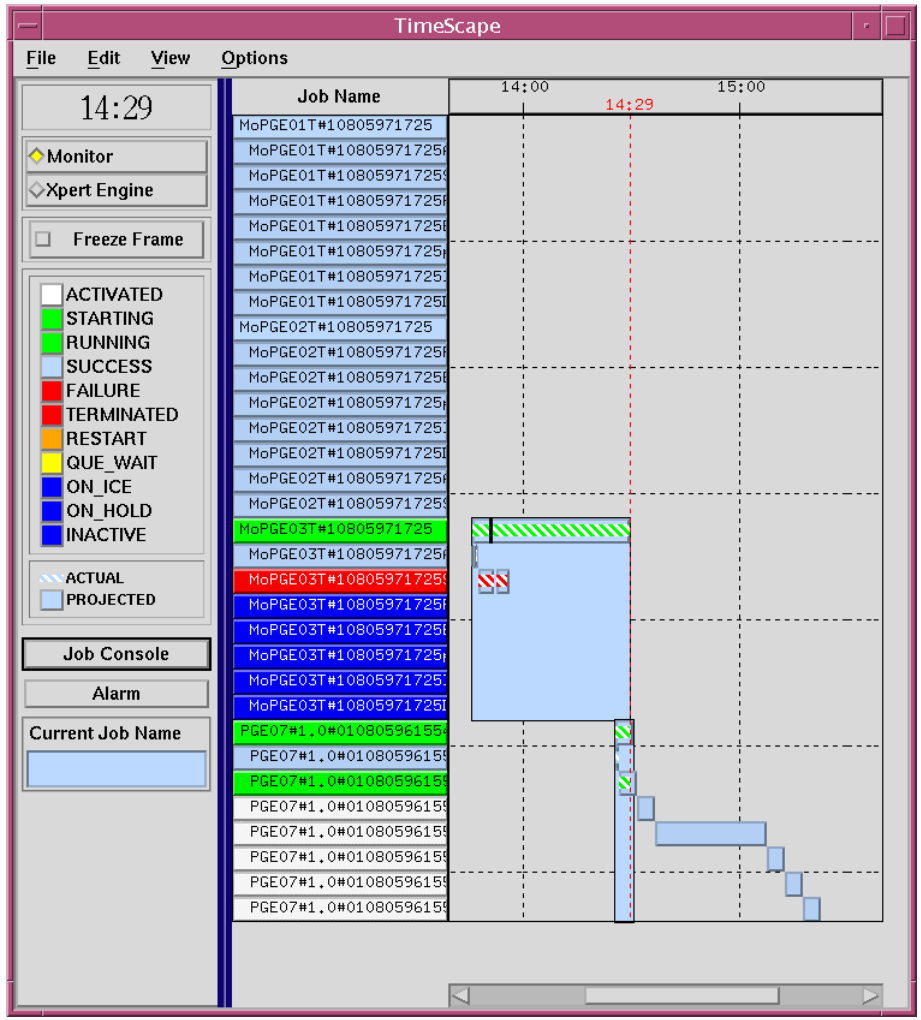
AutoSys GUI Control Panel



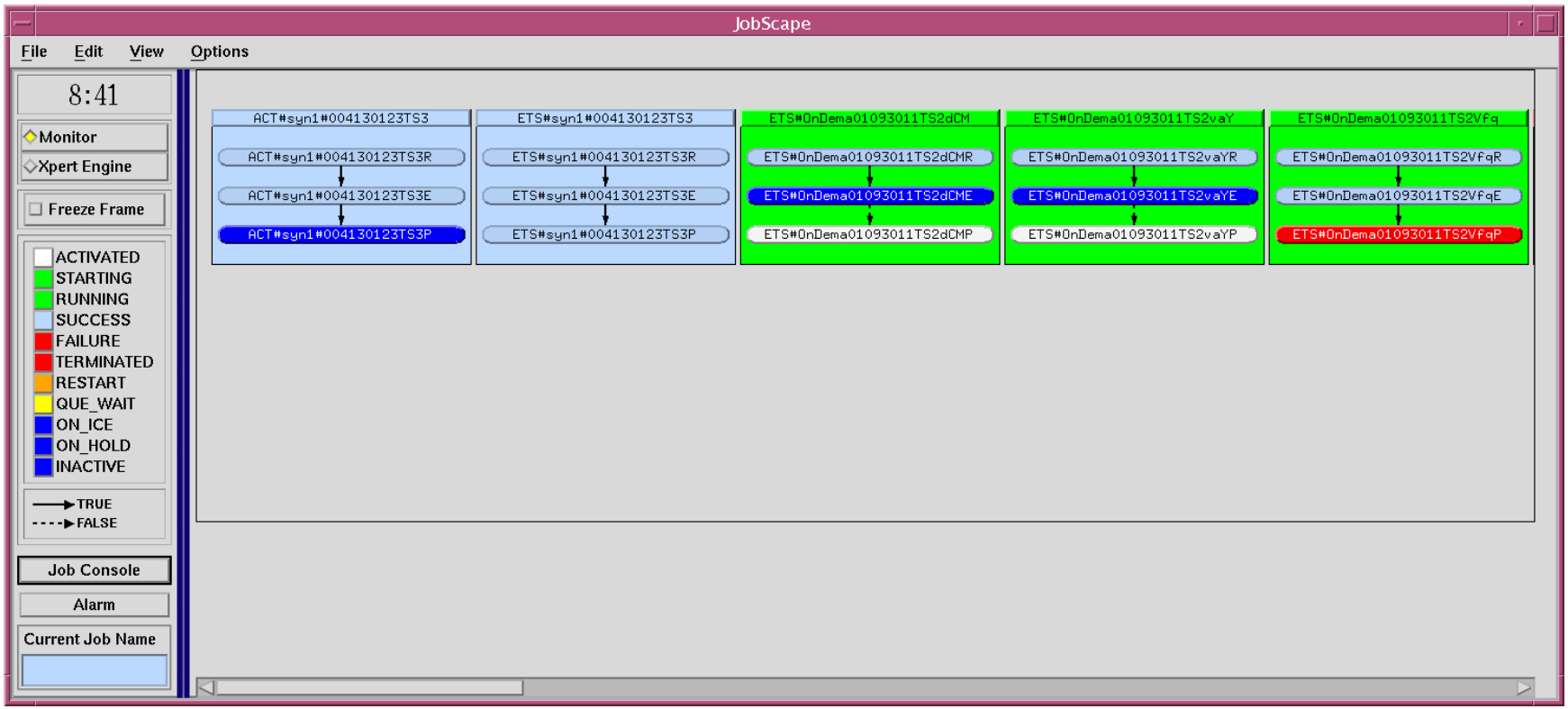
AutoXpert HostScape GUI



AutoXpert TimeScape GUI



AutoXpert JobScape GUI



AutoXpert JobScape GUI Runtime Options Dialogue Box

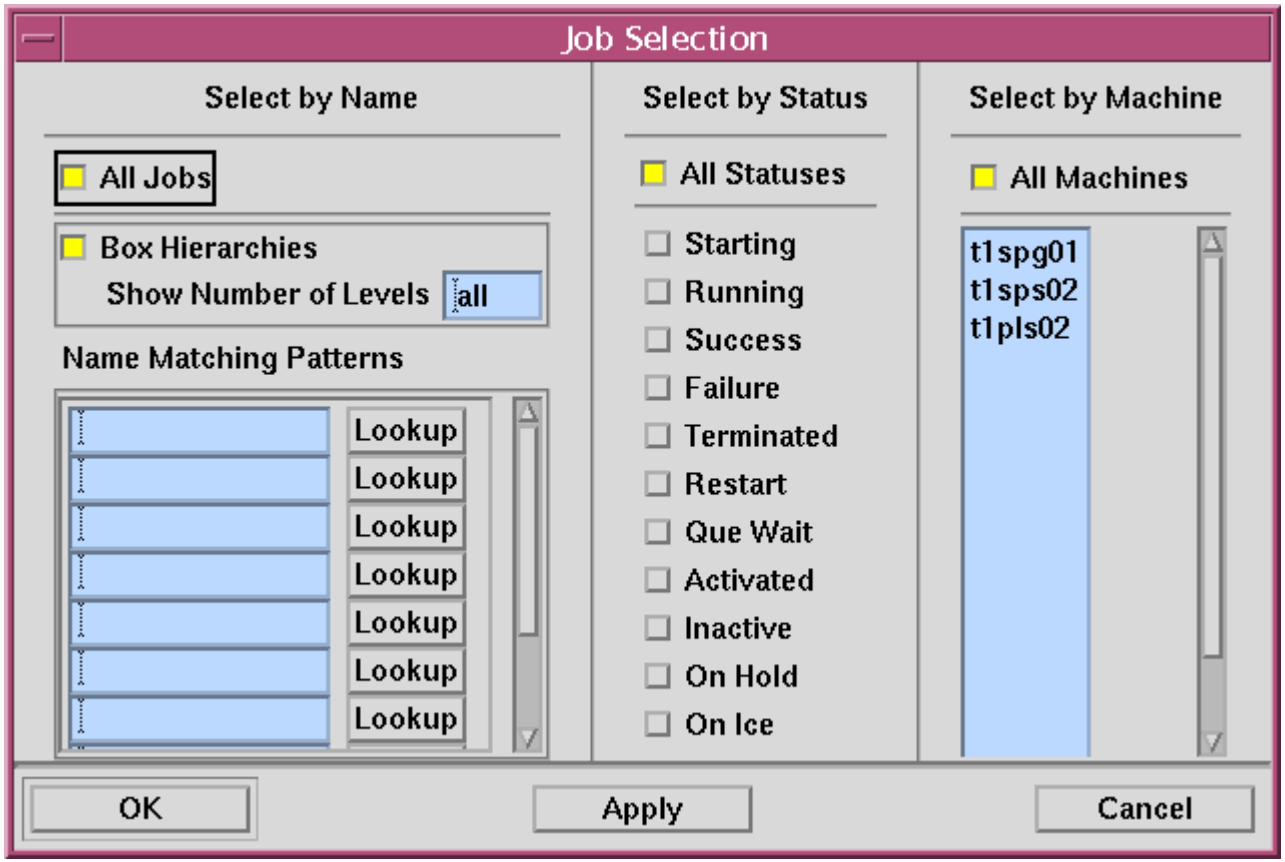
Configuring AutoSys (Cont.): Selecting Jobs to be Displayed

- **Jobs can be selected on the basis of the following criteria:**
  - **Job Name**
  - **Job Status**
  - **Machine**
- **Default values**
  - **All Jobs**
  - **All Statuses**
  - **All Machines**

Configuring AutoSys (Cont.): Selecting Jobs to be Displayed

- **Procedure**
  - Select an AutoXpert GUI
  - Select View → Select Jobs to Display
  - For the Select by Name option select all jobs by clicking on the All Jobs button; to select jobs by pattern matching, enter the appropriate characters/wild cards
  - To select all job statuses click on the All Statuses button; to select jobs by status click on the appropriate Select by Status toggle buttons
  - To select jobs on all machines click on the All Machines button; to select jobs by machine click on the applicable machine name(s) in the Select by Machine area
  - Select OK (or Apply)

Job Selection Dialogue Box



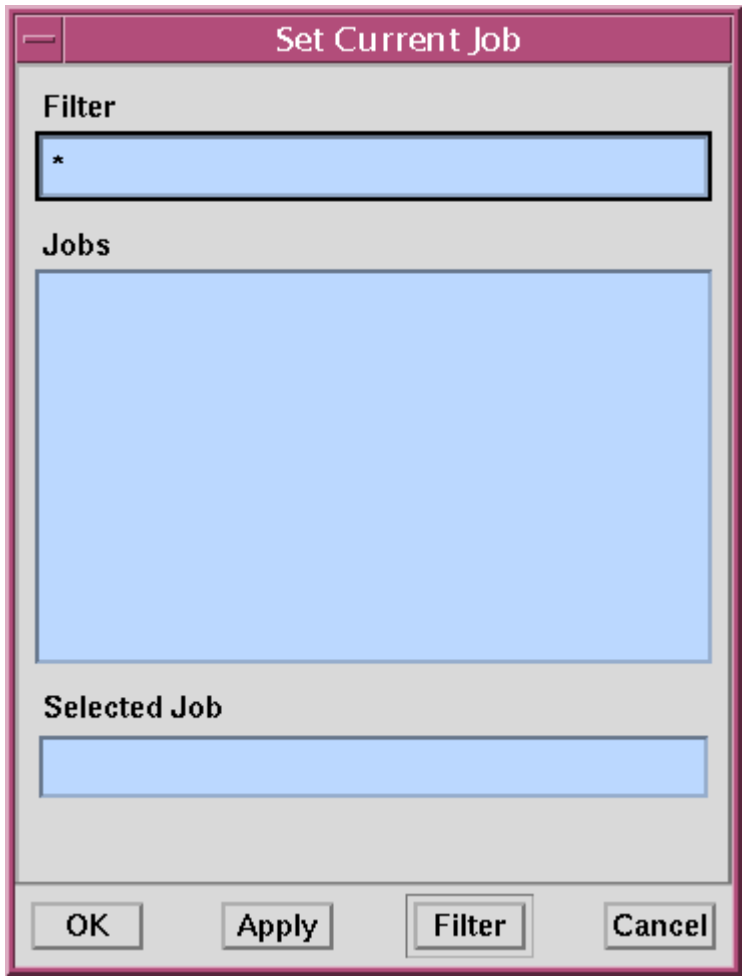
Configuring AutoSys (Cont.): Setting the Current Job

- Setting the current job causes the job name to be displayed in the Current Job Name field in the Control Region of the AutoXpert GUI
  - Subsequently clicking on the Job Console button on the AutoXpert GUI causes the Job Activity Console GUI (also known as the Ops Console GUI) to be displayed with information concerning the current job
- There are two methods for setting the current job:
  - Click on the name of a job displayed on an AutoXpert GUI
  - Set the current job using the AutoXpert GUI pull-down menu

Configuring AutoSys (Cont.): Setting the Current Job

- **Procedure**
  - Select an AutoXpert GUI
  - Select View → Set Current Job
  - Type the *jobname* or a portion of the *jobname* in the Filter field
    - » The asterisk (*) wildcard character can be used for entering a partial job name
  - Click on the Filter button
  - Select (highlight) the job to be designated the "current job" from the jobs listed in the Jobs field
  - Select OK (or Apply)

Set Current Job Dialogue Box



Configuring AutoSys (Cont.)

- **Configuring Hardware Groups**
  - **makes it easier to monitor the hardware associated with a particular function (e.g., testing, training, or a certain type of processing)**
  - **default group is "All Machines"**
  - **Production Monitor defines a specific set of machines to be monitored as a group**
    - » **must know which machines are to be included in the group**
    - » **should devise a useful name for the group**

Configuring AutoSys (Cont.): Configuring Hardware Groups

- **Procedure**
  - Access a terminal window logged in to the Queuing Server host
  - Change directory to the subdirectory (e.g., autouser) containing the set-up files
  - If necessary, source the appropriate set-up file
  - Type vi xpert.groups.*AutoSysInstance*
  - Using vi editor commands create/modify hardware groups as necessary
  - Press the Esc key
  - Save the xpert.groups.*AutoSysInstance* file
  - Launch AutoSys/HostScape
  - Select View → Select Machine Group from the pull-down menu

Configuring AutoSys (Cont.): AutoSys Hardware Group File

groupname: Operations

x0pls02

x0sps06

x0spg01

groupname: SSI&T

x0ais01

x0sps06

x0spg01

Reviewing Hardware Status

- **Hardware status is displayed on the HostScape GUI**
- **Production Monitor can determine the following conditions (among others):**
  - **status of processors**
  - **condition of the AutoSys queue**
  - **whether any processors are overloaded while others are idle**
  - **whether there are any system problems**

Reviewing Hardware Status (Cont.)

- **HostScape displays jobs on a machine-by-machine basis**
  - **which AutoSys server/client machines are up and active**
  - **which jobs are running or have recently run on each machine**
- **HostScape can display hardware status in real-time**

Reviewing Hardware Status (Cont.)

- **Procedure (Reviewing Hardware Status)**
  - **Click on the HostScape button on the AutoSys GUI Control Panel**
  - **Review the Control Region to identify the color codes for the status of the machines**
  - **Review the machine type in the View Region**
  - **Review the machine boxes in the View Region to determine the status of individual machines**
  - **Review the Alarm indicating buttons of individual machines in the View Region**
  - **Review the machine connection status in the View Region**

Reviewing Hardware Status (Cont.)

- **Changing the Hardware Status View**
  - **View Options provide three methods of viewing hardware status:**
    - » **normal**
    - » **global**
    - » **zoom**

Reviewing Hardware Status (Cont.)

- **Procedure (Changing the Hardware Status View)**
  - **Select a machine in the View Region by clicking on its name, then select...**
    - » **first Select View Level...**
    - » **then Global View**
  - **Select a specific machine by clicking on its name, then select Zoom in Machine**
    - » **zoom view is displayed**
    - » **table listing relevant data is displayed**
  - **Select Dismiss**
    - » **global view is displayed**
  - **Select View → Select View Level then select Normal View**
    - » **normal view is displayed**

Monitoring/Controlling Job Processing

- There are three primary tools for monitoring and controlling job processing:
  - AutoXpert JobScape GUI
  - AutoXpert TimeScape GUI
  - AutoSys Job Activity Console GUI (also known as also known as the Ops Console GUI)

Monitoring/Controlling Job Processing (Cont.)

- **JobScape GUI**
  - **presents a Pert-like graphical view of job processing from a logical (or job dependency) point of view**
  - **depicts all job types; i.e., command jobs, box jobs, and file-watcher jobs**
  - **depicts the nesting of jobs within boxes and the dependencies between jobs.**
  - **can be used for monitoring and controlling job flow in real-time**
  - **allows the Production Monitor to identify potential problems before they become actual problems, put problem jobs on hold in favor of letting good jobs run, restart jobs after correcting problems with them, etc.**

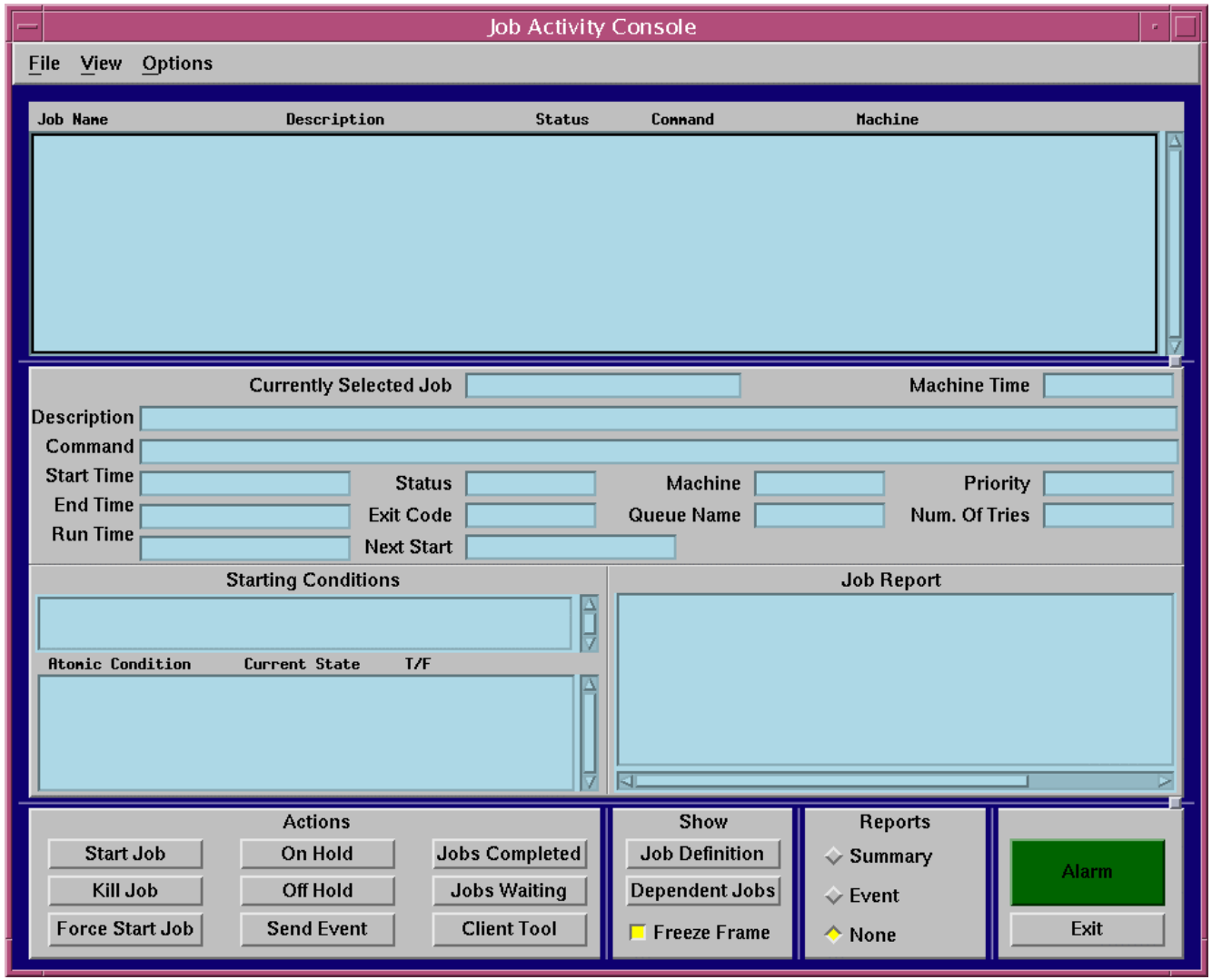
Monitoring/Controlling Job Processing (Cont.)

- **TimeScape GUI**
  - **presents a Gantt-like graphical view of a job processing from a temporal (time-related) point of view**
  - **depicts both command jobs and box jobs**
  - **depicts the nesting of jobs within boxes and the duration of time it will take for jobs to complete**
  - **is used for monitoring job flow in real-time**

Monitoring/Controlling Job Processing (Cont.)

- **Job Activity Console GUI (Ops Console)**
  - **is a text-based interface for monitoring jobs that have been defined for AutoSys**
  - **displays information on the job's start time (and date), end time (and date), run time, status, exit code (if completed), host, priority, and other attributes**
  - **provides a means of evaluating job starting conditions, which can be useful in determining what "upstream" job may be preventing the currently selected job from running**
  - **provides summary and event reports that can be used in identifying problems with processing a particular job**

Job Activity Console (OPS Console)



Monitoring/Controlling Job Processing (Cont.)

- AutoSys defines job status in the terms listed in the table
  - different states are color-coded on JobScape and TimeScape displays

Monitoring/Controlling Job Processing (Cont.)

- **Job States**
  - **Activated (white)**
  - **Starting (green)**
  - **Running (green)**
  - **Success (light blue)**
  - **Failure (red)**
  - **Terminated (red)**
  - **Restart (orange)**
  - **Que_Wait (yellow)**
  - **On_Ice (dark blue)**
  - **On_Hold (dark blue)**
  - **Inactive (dark blue)**

Monitoring/Controlling Job Processing (Cont.)

- **Production Monitor may need to modify a particular job in any of the following ways:**
  - **Start the job**
  - **Kill the job**
  - **Force the job to start**
  - **Place the job on hold**
  - **Take the job off hold**
- **Three methods for making those types of modifications to a particular job:**
  - **Buttons in the Actions region of the Job Activity Console (Ops Console)**
  - **Menu accessed by clicking the right mouse button on the relevant job name on JobScape or TimeScape**
  - **AutoSys Send Event GUI**

Monitoring/Controlling Job Processing (Cont.)

- In AutoSys terms a control action such as starting or killing a job is accomplished by sending an "event" to the job
  - An event is basically a message
  - For example:
    - » Clicking on the Start Job button on the AutoSys Job Activity Console begins the process by which AutoSys sends a "start" message to the Currently Selected Job
- In addition to modifications to job status, the buttons in the Actions region of the Job Activity Console (Ops Console) allow the Production Monitor to generate reports:
  - Jobs Completed
  - Jobs Waiting

Jobs Completed Report

emacs@tlsps02.vatc.ecs.nasa.gov

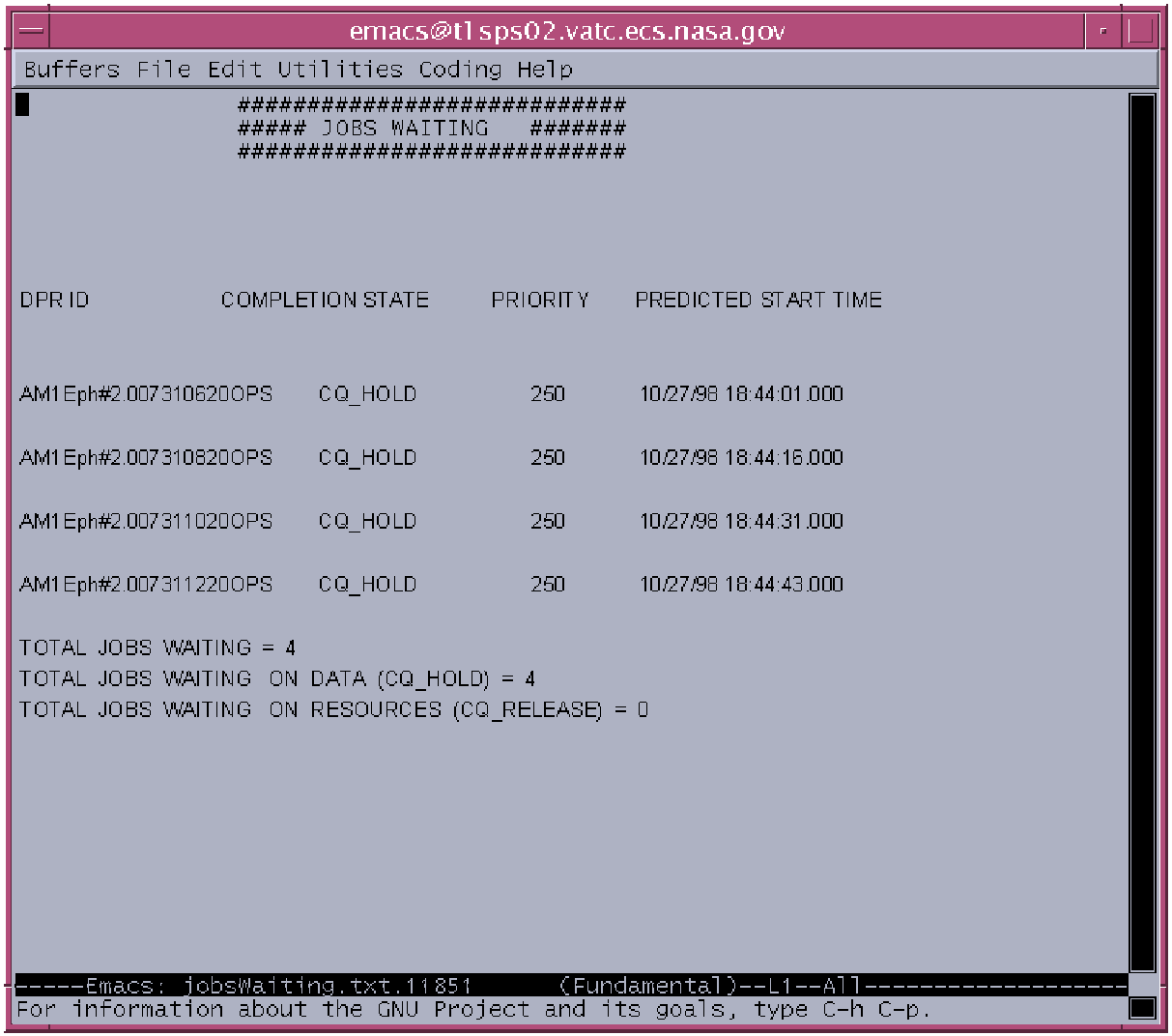
Buffers File Edit Utilities Coding Help

```
                ############################
                ##### JOBS COMPLETED #######
                ############################


DPR ID                  COMPLETION STATE   PRIORITY  DATA START TIME    AUTOSYS

MISRLIKE#s03050000TS    SUCCESS             2750  12/03/93 00:00:00.000   VAT
BTS#OnDema01093011TS    SUCCESS             120   04/01/95 04:30:11.000   VAT
ACT#syn1#004130123TS    SUCCESS             2750  07/04/97 09:01:23.000   VAT
BTS#syn1#004130123TS    SUCCESS             2750  07/04/97 09:01:23.000   VAT
FddAtt#30112312200TS    SUCCESS             2750  12/31/99 17:00:00.000   VAT
AM1Eph#30101010000TS    CANCELED            2750  12/31/99 19:00:00.000   VAT
FddAtt#30101010000TS    SUCCESS             2750  12/31/99 19:00:00.000   VAT
MoPGE01#sy01010000TS    CANCELED            2750  12/31/99 19:00:00.000   VAT
MoPGE02#sy01000500TS    CANCELED            2750  12/31/99 19:05:00.000   VAT
MoPGE03#sy01000500TS    CANCELED            2750  12/31/99 19:05:00.000   VAT
MoPGE02#sy01001000TS    CANCELED            2750  12/31/99 19:10:00.000   VAT
MoPGE03#sy01001000TS    CANCELED            2750  12/31/99 19:10:00.000   VAT
MoPGE02#sy01001500TS    CANCELED            2750  12/31/99 19:15:00.000   VAT
MoPGE03#sy01001500TS    CANCELED            2750  12/31/99 19:15:00.000   VAT
MoPGE02#sy01002000TS    CANCELED            2750  12/31/99 19:20:00.000   VAT
```

-----Emacs: jobsCompleted.txt.11451 (Fundamental)--L1--Top----------

For information about the GNU Project and its goals, type C-h C-p.

Jobs Waiting Report



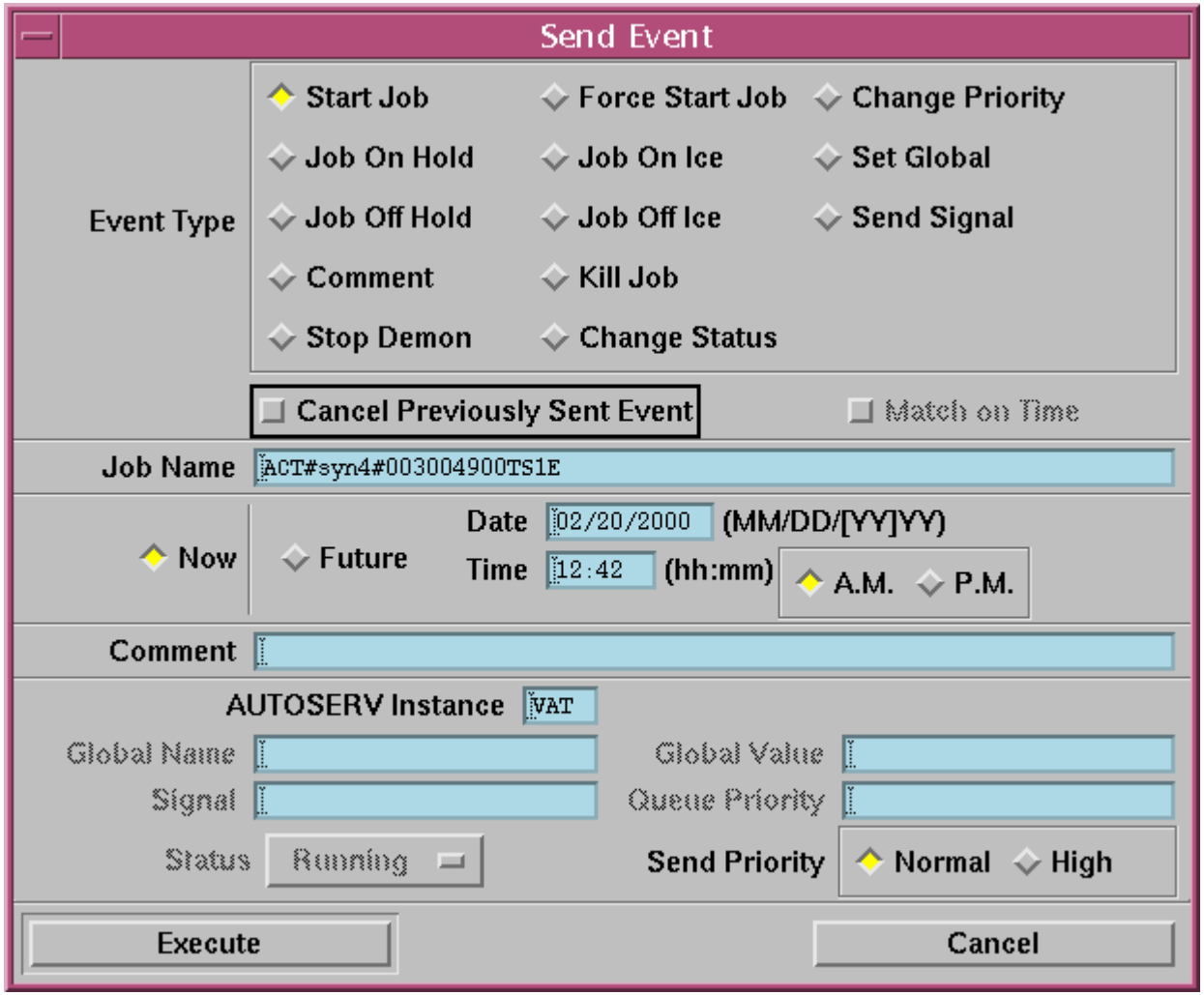
Monitoring/Controlling Job Processing (Cont.)

- The menu accessed using the right mouse button on one of the AutoXpert GUIs allows the Production Monitor to initiate either of the following actions (in addition to the previously mentioned modifications to job status):
  - Put the job on ice
  - Take the job off ice

Monitoring/Controlling Job Processing (Cont.)

- **Send Event GUI allows the Production Monitor to initiate any of the following actions:**
  - **Start the job**
  - **Kill the job**
  - **Force the job to start**
  - **Place the job on hold**
  - **Take the job off hold**
  - **Change the job's status**
  - **Change the job's priority**
  - **Put the job on ice**
  - **Take the job off ice**
  - **Stop the daemon (stop the Event Processor in an emergency)**
  - **Set a global value**
  - **Send a signal concerning the job**
  - **Make a comment (for example, why a job start was forced)**

Send Event GUI



Monitoring/Controlling Job Processing (Cont.)

- **Guidelines for Reporting Unsuccessful Completion of On-Demand Jobs**
  - **Under any of the following circumstances involving an on-demand job notify User Services of the problem in accordance with the applicable local policy**
    - » **Job is killed**
    - » **Job terminates and cannot be restarted**
    - » **A FAILPGE granule is created**
  - **The DAAC is obliged to send an e-mail message to the requester of an unsuccessful on-demand job to explain why the request cannot be fulfilled**

Monitoring/Controlling Job Processing (Cont.)

- **Guideline for Putting Jobs "On Ice" or "On Hold"**
  - **Ensure that the job to be put either "on hold" or "on ice" is not already in a "starting" or "running" state**
    - » **A job that is either "starting" or "running" cannot be put "on hold" or "on ice"**

Monitoring/Controlling Job Processing (Cont.)

- **Guidelines for Force-Starting Jobs**
  - **Force-start command jobs (e.g., preprocessing or preprocessing) only**
  - **Do not attempt to force-start a box job**
    - » **The software does not support box job force-starts**
    - » **Force-starting a box job can cause the PDPS database to get out of sync and prevent the DPR (and possibly other DPRs) from running successfully**
    - » **If a box job were force-started, the allocation portion of the preprocessing job job would run again and might choose a different science processor than was chosen the previous time the job ran, which could cause failure of the job**
    - » **Box job force-starts lack the code needed to check the state of the box and perform the database cleanup activities necessary for starting over**

Monitoring/Controlling Job Processing (Cont.)

- **Guidelines for Force-Starting Jobs (Cont.)**
  - **Ensure that the GUI has refreshed and the job to be force-started is not already running before trying to force-start a job**
    - » **If a job is already running, it should not be force-started**
    - » **If using AutoSys/AutoXpert 3.4.2 or a later version, it should not be possible to force-start jobs that are already running**
  - **If any command job other than execution fails, force-start the job that failed only**
  - **Do not force start any preceding or succeeding jobs in the box**

Monitoring/Controlling Job Processing (Cont.)

- **Guidelines for Force-Starting Jobs (Cont.)**
  - **If execution fails, it is not safe to restart it unless the post-processing job had been put on hold and the failure was detected before post-processing started running**
  - **If execution fails and the failure was not detected before post-processing started running, the DPR must run to completion as a failed PGE and the DPR must be deleted and recreated**
  - **In any case the Production Monitor may implement certain changes of job status only when the Production Monitor "owns" the job affected by the modification**

Monitoring/Controlling Job Processing (Cont.)

- **Procedure (Monitoring/Controlling Job Processing)**
  - **Click on AutoSys GUI Control Panel buttons to display GUIs (as necessary)**
    - » **JobScape**
    - » **TimeScape**
    - » **Job Activity Console (Ops Console)**
  - **Configure AutoSys/AutoXpert runtime options for the GUIs as necessary**
  - **If necessary, select jobs to be displayed on the GUIs**
  - **Observe information displayed on the GUI(s)**
  - **Perform subordinate procedures as necessary**
  - **Repeat preceding steps as necessary to monitor/control jobs**

Monitoring/Controlling Job Processing (Cont.)

Good vs Bad

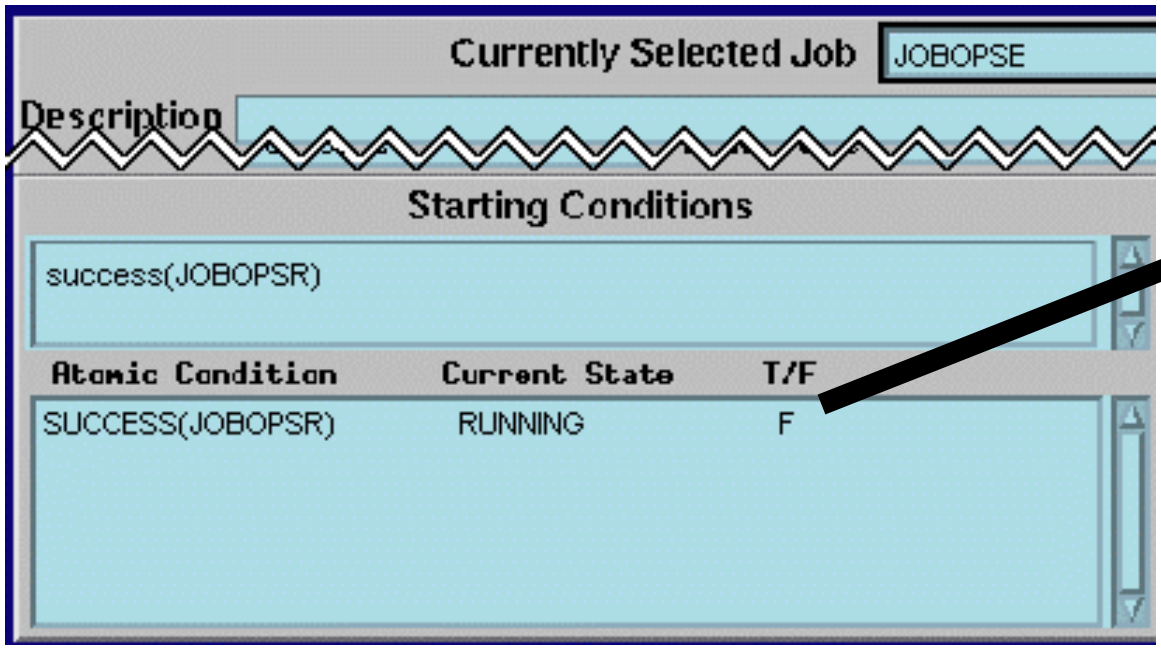
Looks Good!

Trouble!

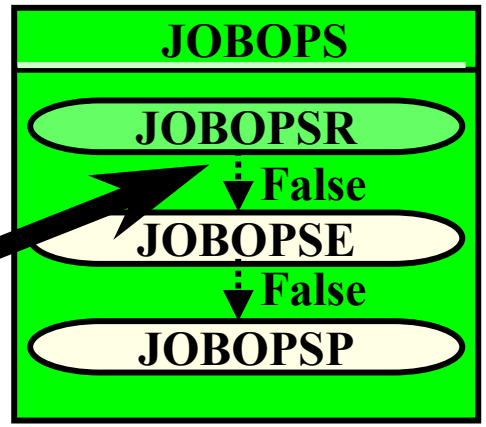
Monitoring/Controlling Job Processing (Cont.)

Nuke'm

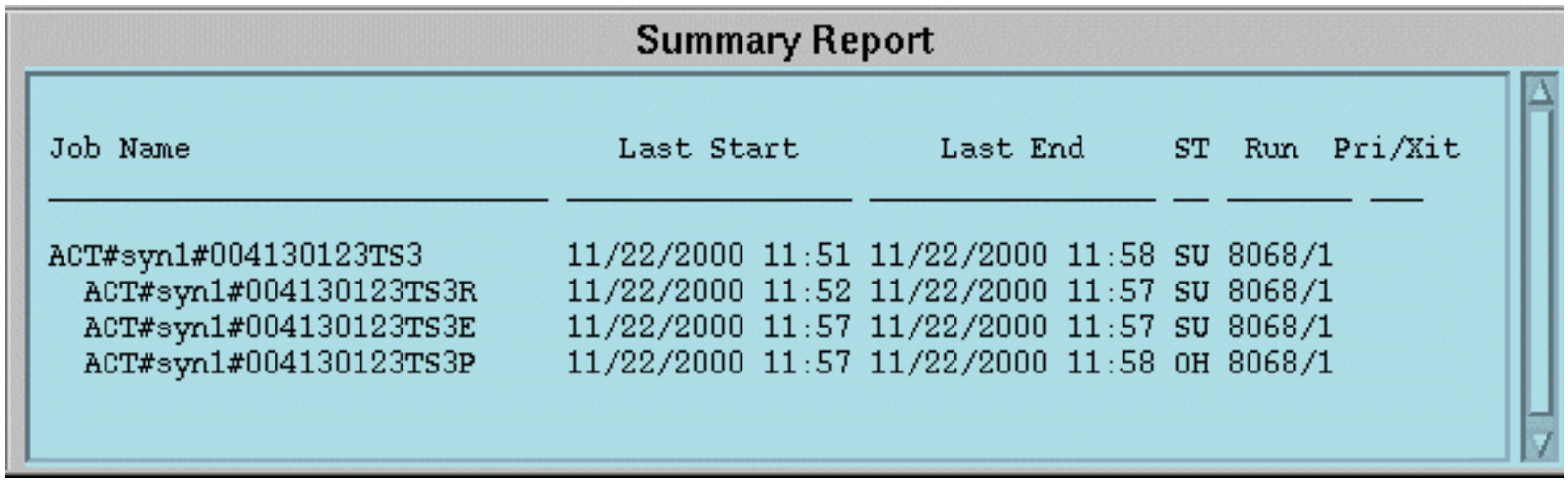
Job Console



JobScape GUI



Example of A Summary Report on the Job Activity Console



Monitoring/Controlling Job Processing (Cont.)

- **Procedure (Determining the Descendants of a Job)**
  - **Select the job by placing the mouse cursor on the job and clicking with the left mouse button**
  - **Place the mouse cursor on the job and click and hold the right mouse button**
  - **If applicable, select (highlight) Show Children from the pop-up menu (release the right mouse button)**
  - **If applicable, select Show All Descendants from the pop-up menu**
  - **If applicable, select Hide All Descendants from the pop-up menu**

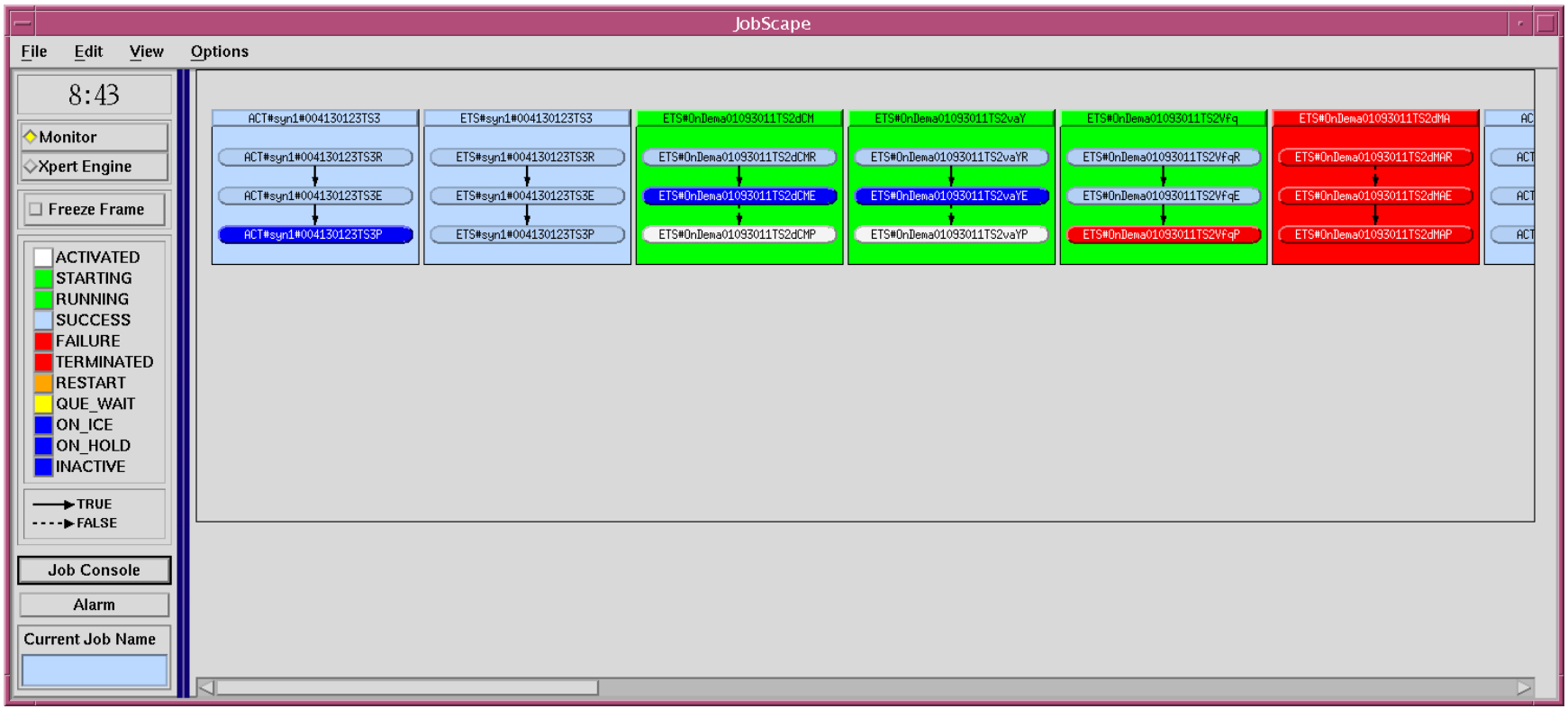
Monitoring/Controlling Job Processing (Cont.)

- **Changing the JobScape View Using the Pull-Down Menu**
  - **Changing the view affects the level of detail displayed in the View Region of the GUI**
  - **The view can be changed in two ways**
    - » **Simply clicking with the right mouse button on the name of a job displayed on an AutoXpert GUI and selecting the desired option from the pop-up menu**
    - » **Using the View pull-down menu (has some additional options)**

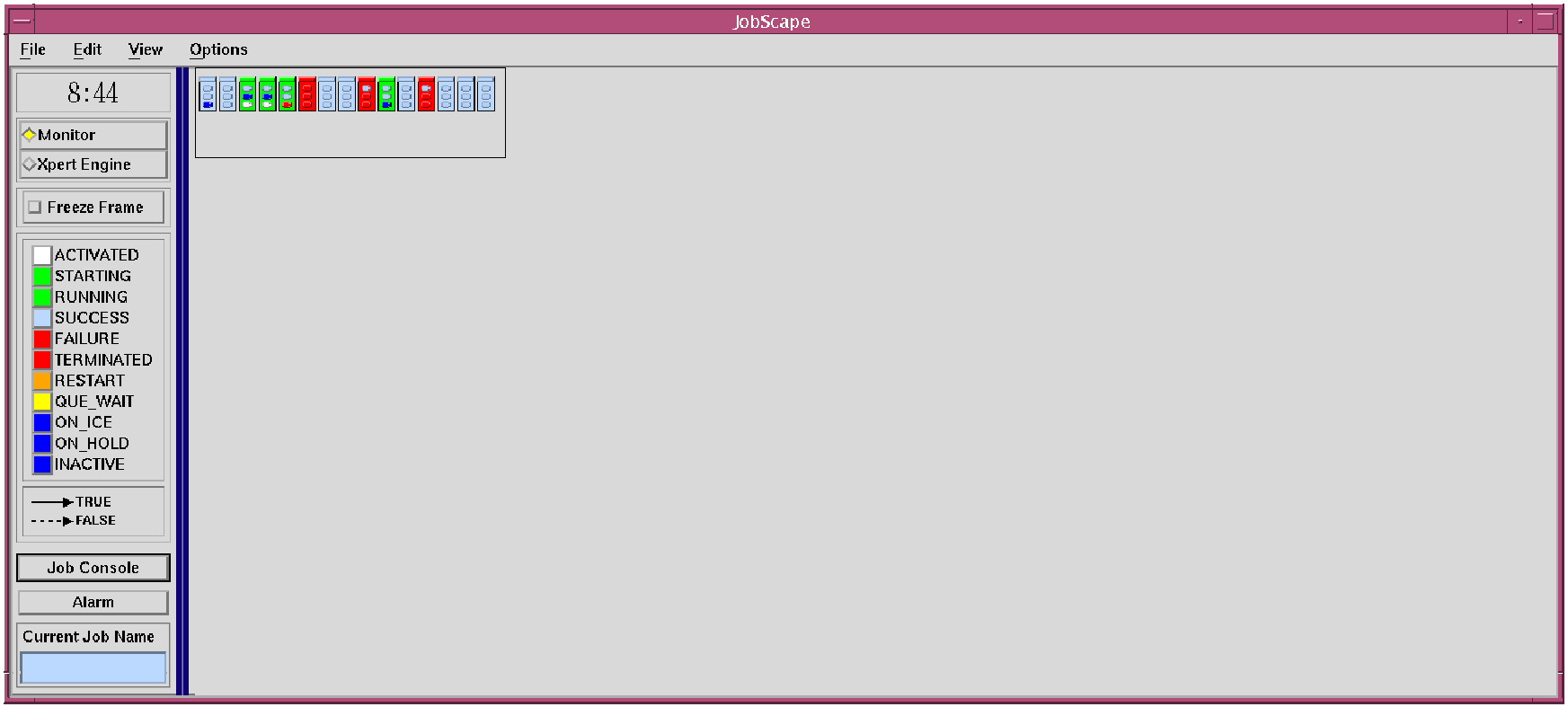
Monitoring/Controlling Job Processing (Cont.)

- **Procedure (Changing the JobScape View Using the Pull-Down Menu)**
  - **Select View → Set View from the pull-down menu**
  - **Select the desired option from the pull-down menu**
    - **Normal Text View**
    - **Small Text View**
    - **No Text View**
    - **Show Arrows**
    - **Hide Arrows**
    - **View by Id**
  - **Select View → Set Display Levels**
  - **Select the desired option from the pull-down menu**
    - **All is the default type of view**

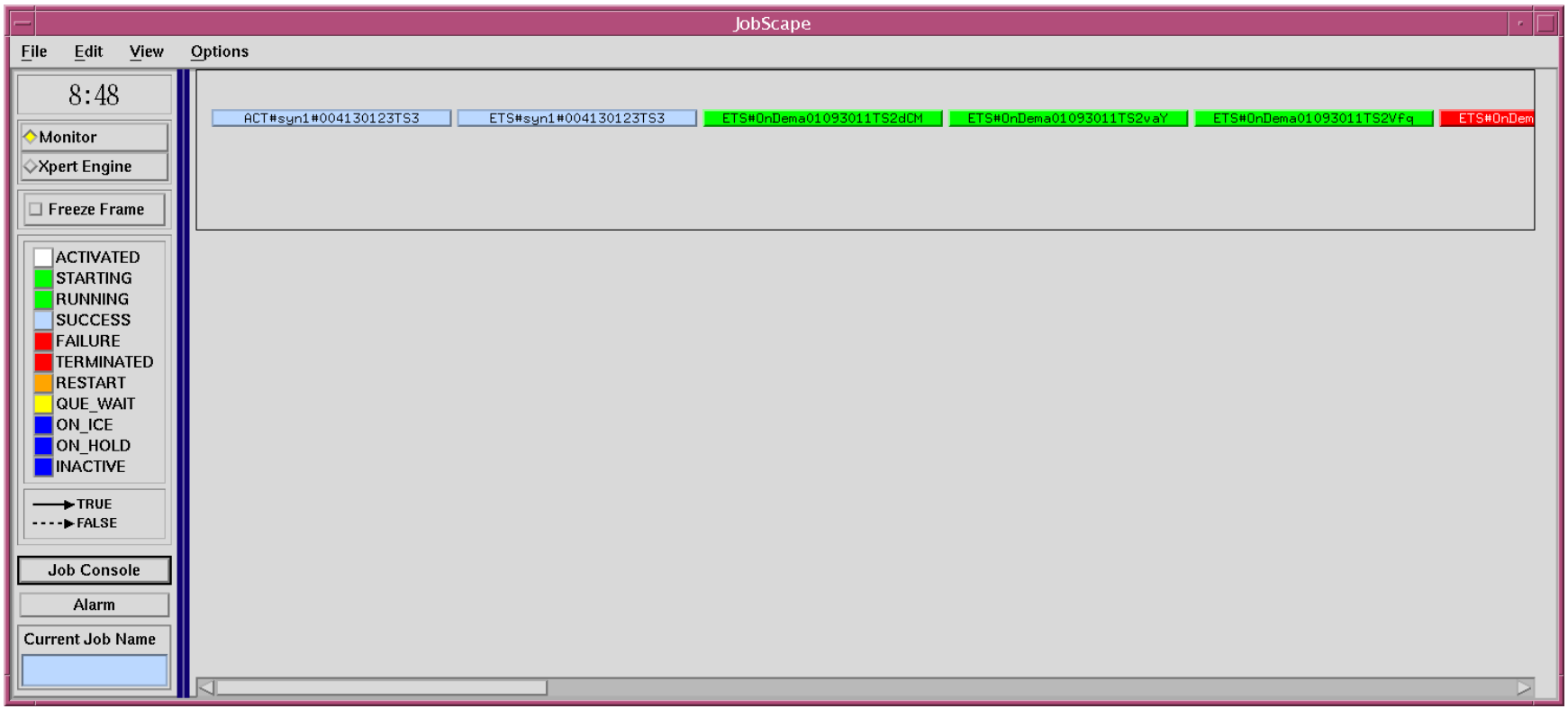
JobScape GUI “Small Text” View



JobScape GUI “No Text” View



JobScape GUI “One Level” View



Monitoring/Controlling Job Processing (Cont.)

- **Reviewing Alarms**
  - **Alarms indicate problems with job processing**
    - » **failure of job processing**
    - » **database problem**
    - » **communication problem**
    - » **hardware or software failure**
    - » **some other error in the data processing system**
  - **Production Monitor reviews alarms using the AutoSys Alarm Manager**
    - » **view alarms as they arrive**
    - » **provide a response to an alarm**
    - » **change alarm status**

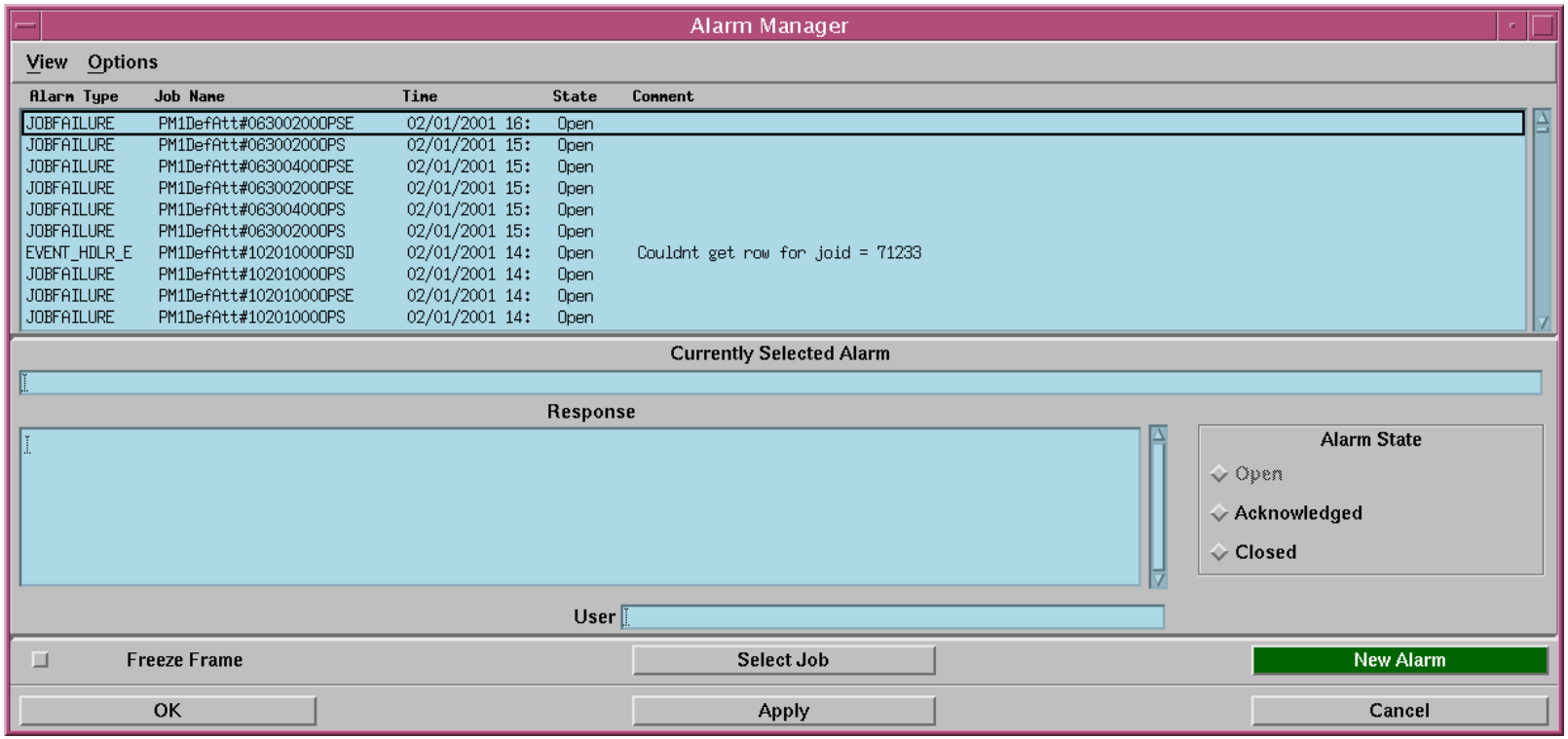
Monitoring/Controlling Job Processing (Cont.)

- **Reviewing Alarms (Cont.)**
  - **Production Monitor can configure the Alarm Manager to display certain types of alarms only**
    - » **type of alarm**
    - » **alarm state**
    - » **time of the alarm**
  - **For descriptions of AutoSys alarms refer to the AutoSys® Reference Guide for UNIX**
    - » **The AutoSys® Reference Guide for UNIX, the AutoSys® User Guide for UNIX, and the AutoSys®/Xpert User Guide for UNIX can be downloaded from http://support.ca.com/autosysmanuals/atsys.html**

Monitoring/Controlling Job Processing (Cont.)

- **Procedure (Reviewing Alarms)**
  - Click on the Alarm button to display the Alarm Manager GUI
  - Configure alarm selection as described in the procedure for Configuring Alarm Selection
  - Click on an alarm in the Alarm List
  - Click the Response edit box and type in a response, if desired
  - Update the Alarm State by clicking on the appropriate radio button
  - Click on the Apply button
  - Repeat steps as necessary to review and update multiple alarms
  - Click on the OK button to dismiss the Alarm Manager GUI

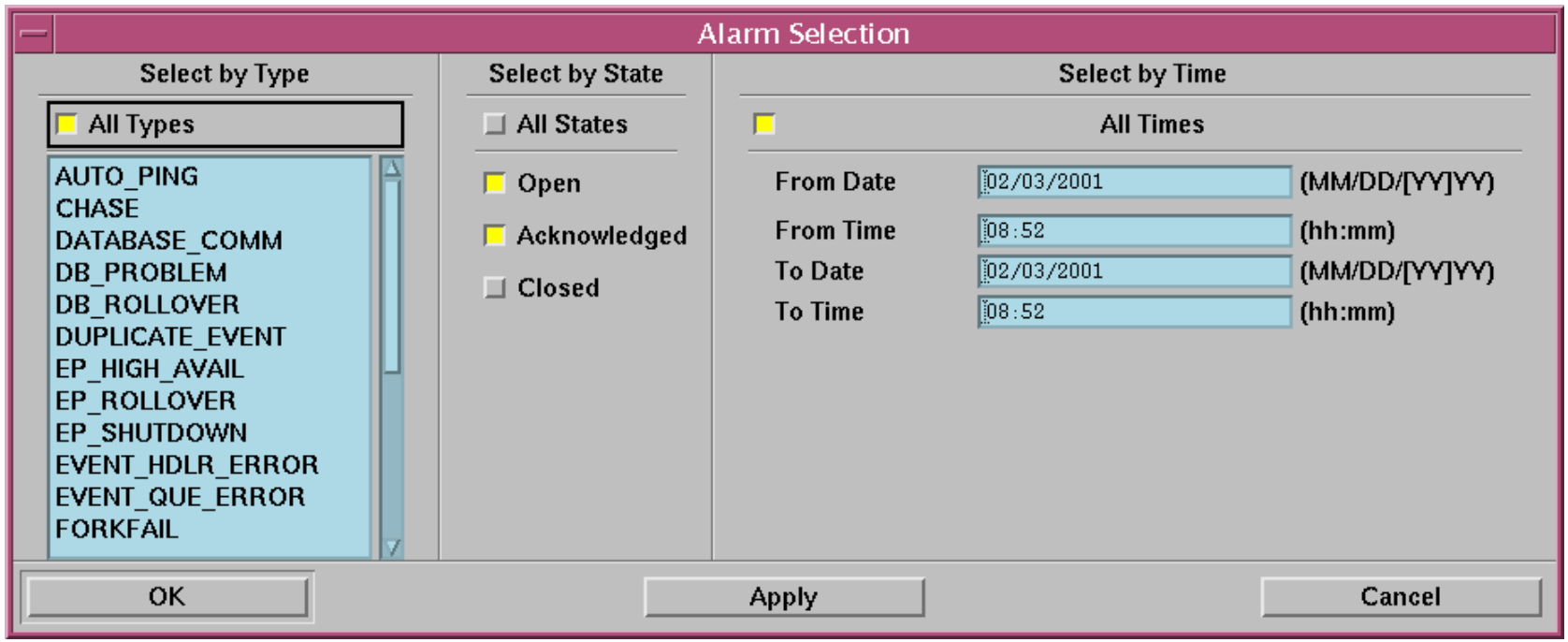
Alarm Manager GUI



Monitoring/Controlling Job Processing (Cont.)

- **Procedure (Configuring Alarm Selection)**
  - **Select View → Select Alarms from the pull-down menu to display the Alarm Selection GUI**
  - **Click on the desired alarm(s) in the Select by Type alarm list; to select all types of alarms, click on the All Types button**
  - **Click on the All States button to select all alarm states; to select alarms by state click on whichever of the Select by State toggle buttons properly describe(s) the state(s) to be selected**
  - **Click on the All Times button to select all times; to select alarms by time type the starting date/time and ending date/time in the applicable fields**
  - **Apply selections**

Alarm Selection GUI



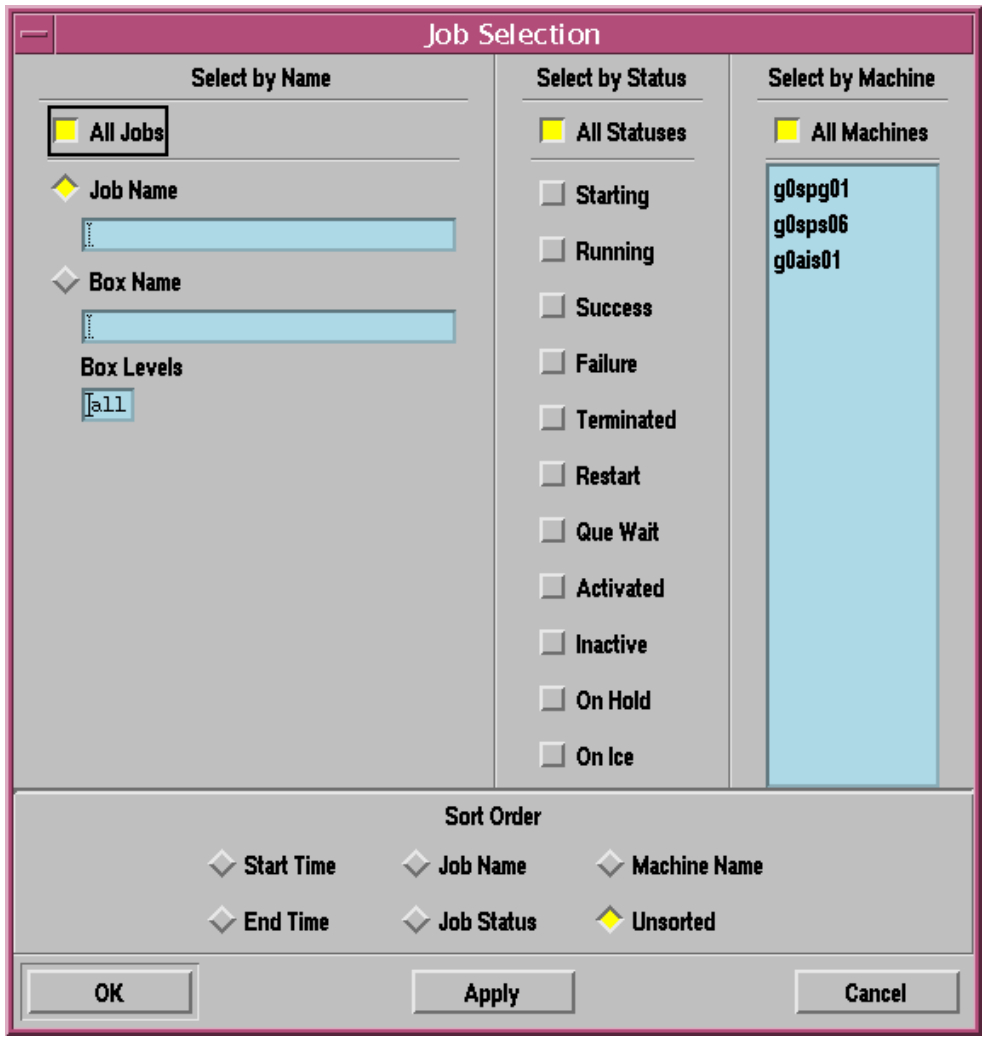
Monitoring/Controlling Job Processing (Cont.)

- **Specifying Job Selection Criteria**
  - **Production Monitor reviews job activities using the AutoSys Job Activity Console**
  - **AutoSys Job Selection GUI is used for...**
    - » **specifying (filtering) jobs the Production Monitor will review**
    - » **setting the criteria for displaying jobs by name, status and/or machine**

Monitoring/Controlling Job Processing (Cont.)

- **Procedure (Specifying Job Selection Criteria)**
  - Click on the Ops Console button on the AutoSys GUI Control Panel
  - Choose View → Select Jobs from the pull-down menu to display the Job Selection GUI
  - For the Select by Name option select all jobs by clicking on the All Jobs button
  - To select all job statuses click on the All Statuses button; to select jobs by status click on the appropriate Select by Status toggle buttons
  - To select all machines click on the All Machines button; to select individual machines click on the machines in the list in the Select by Machine area
  - Click on the desired order in the Sort Order area
  - Apply the selections

Job Selection GUI



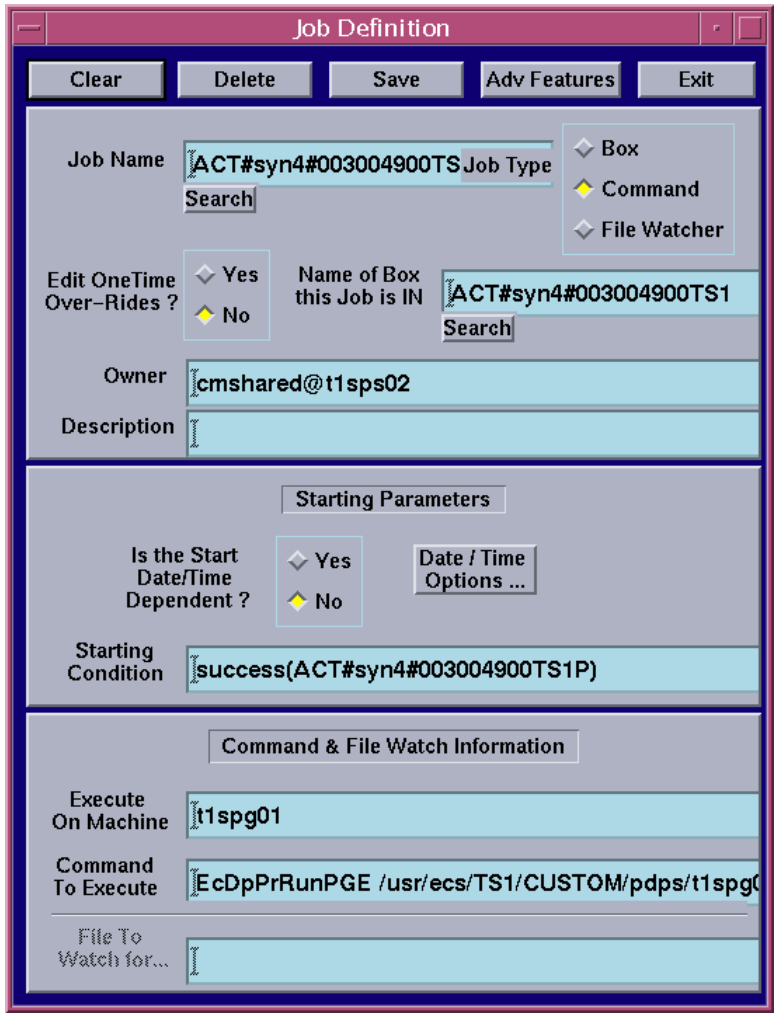
Monitoring/Controlling Job Processing (Cont.)

- **Determining the Ownership of an AutoSys Job**
  - **AutoSys is very much ownership-aware**
    - **Only the "owner" of a job has "edit" privileges and can make changes to the status of an owned job**
  - **AutoSys recognizes ownership in terms of two factors:**
    - **UserID**
    - **Machine where the operator (user) logged in**
  - **Example:**
    - **cmshared@g0sps06 identifies the Production Monitor who logged in as "cmshared" at g0sps06**
    - **Any operator who logs in as "cmshared" at another machine (e.g., g0pls01) would not be able to change the status of a job "owned" by cmshared@g0sps06**
    - **To have any real effect on a job first it is necessary to log in as the job's owner and launch AutoSys GUIs**

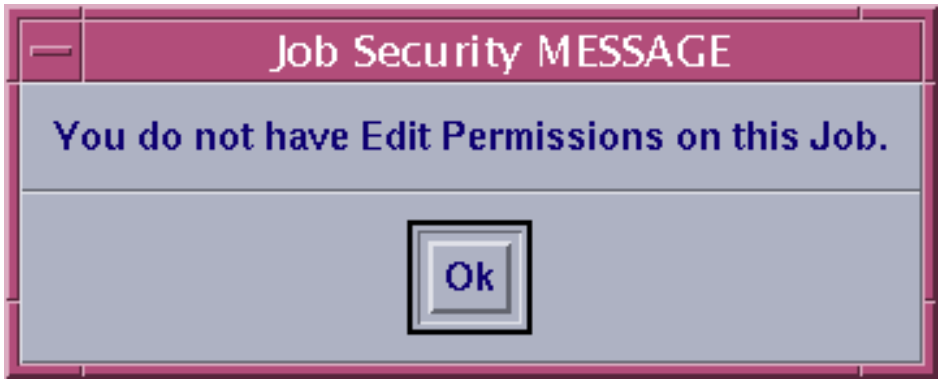
Monitoring/Controlling Job Processing (Cont.)

- **Procedure (Determining the Ownership of a Job)**
  - **Click on the JobScape button (or TimeScape button) on the AutoSys GUI Control Panel**
  - **Place the mouse cursor on the relevant job and click and hold the right mouse button**
  - **Select (highlight) Job Definition from the pop-up menu (release the right mouse button)**
  - **Review the entry in the Owner field of the Job Definition GUI**
  - **To exit from the Job Definition GUI, click on the Exit button**

Job Definition GUI



Job Definition GUI
Job Security MESSAGE Window



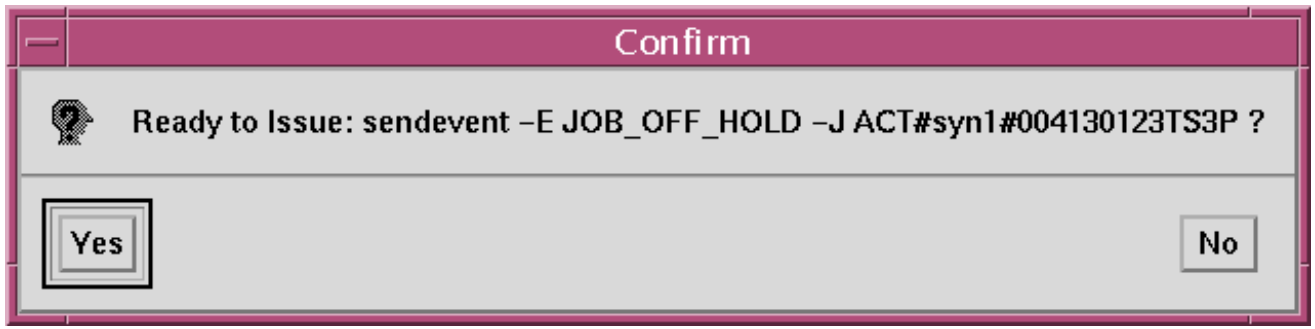
Monitoring/Controlling Job Processing (Cont.)

- **Sending an Event to a Job**
  - **As previously mentioned there are three methods for making certain types of modifications (e.g., start or kill) to a particular job**
    - » **Menu accessed by clicking the right mouse button on the relevant job name on either the JobScape or TimeScape GUI**
    - » **Buttons in the Actions region of the Job Activity Console (Ops Console)**
    - » **AutoSys Send Event GUI**

Monitoring/Controlling Job Processing (Cont.)

- **Procedure (Sending an Event to a Job from an AutoXpert GUI)**
  - **Select either JobScape or TimeScape from the AutoSys GUI Control Panel**
  - **Place the mouse cursor on the relevant job and click and hold the right mouse button**
  - **Select the event (e.g., Force Start Job) to be sent to the job from the pop-up menu**
    - » **If there is no option corresponding to the desired action, modify job status from either the Job Activity Console or the Send Event GUI**
  - **Select Yes to confirm sending the event**

Confirmation Dialogue Box



Monitoring/Controlling Job Processing (Cont.)

- **Procedure (Sending an Event to a Job from the Job Activity Console)**
  - **Select jobs for display on the Job Activity Console (Ops Console)**
  - **Verify that the job with the status to be modified is listed in the Currently Selected Job field of the Job Activity Console**
  - **Click on the button corresponding to the desired action to be taken with respect to the selected job**
    - » **If there is no button corresponding to the desired action, modify job status using the Send Event GUI**
  - **Click on Yes to send the event to the job**

Monitoring/Controlling Job Processing (Cont.)

- **Sending an Event to a Job from the Send Event GUI**

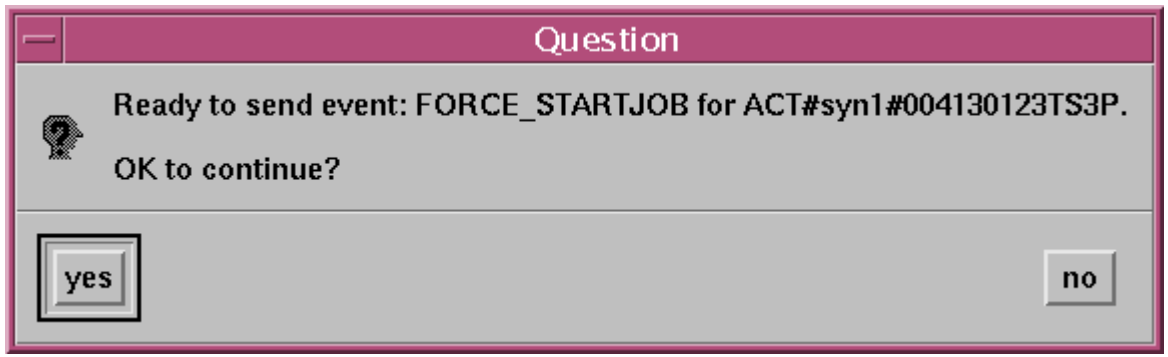
CAUTION

Once an event has been sent from the Send Event dialog, it may not be possible to cancel or modify it.

Monitoring/Controlling Job Processing (Cont.)

- **Procedure (Sending an Event to a Job from the Send Event GUI)**
  - **Click on the Ops Console button and list jobs**
  - **In the Job List region of the Job Activity Console click on the job row corresponding to the job**
  - **Click on the Send Event button**
  - **Select the Event Type**
  - **Enter the desired date and time when the job status is to be modified**
  - **Change the Queue Priority entry if Change Priority was selected**
  - **Select Status to send if Change Status was selected**
  - **Select the Send Priority**
  - **Click on the Execute button**

Send Event GUI: Confirmation Dialogue Box



Monitoring/Controlling Job Processing (Cont.)

- **Procedure (Canceling a Sent Event)**
  - Click on the Send Event button in the Actions Region of the Job Activity Console
  - Click on Event Type to select the type of event that was sent to the job and is to be canceled
  - Click on the Cancel Previously Sent Event button
  - Verify that the correct job is listed in the Job Name field of the Send Event GUI
  - Click on the Execute button
  - Click on Yes to confirm canceling the sent event

Monitoring/Controlling Job Processing (Cont.)

- **Performing Job Management Client Functions**
  - **Job Management Client tool is a set of utility programs intended primarily for use by software developers**
  - **Get access to any of the following Job Management Client functions from AutoSys by clicking on the Client Tool button in the Actions region of the Job Activity Console**
    - » **Create DPR Job**
    - » **Release DPR Job**
    - » **Cancel DPR Job**
    - » **Change DPR ID**
    - » **View Job Management DPR Queue**
    - » **Create Ground Event Job**
    - » **Cancel Ground Event Job**

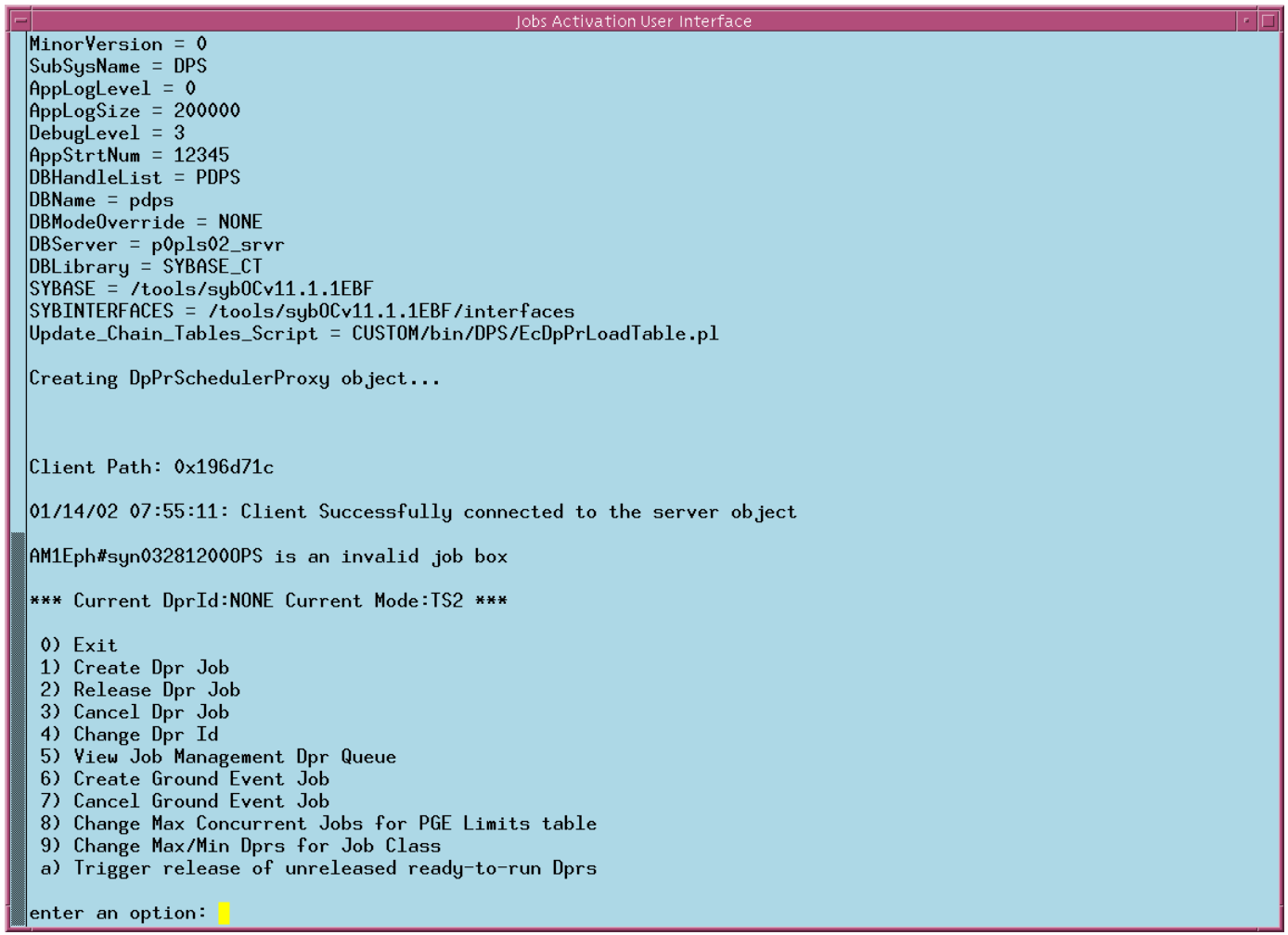
Monitoring/Controlling Job Processing (Cont.)

- **Performing Job Management Client Functions (Cont.)**
  - **Get access to Job Management Client functions (Cont.)**
    - » **Change Max Concurrent Jobs for PGE Limits table**
    - » **Cancel Max/Min Dprs for Job Class**
    - » **Trigger release of unreleased ready-to-run Dprs**

Monitoring/Controlling Job Processing (Cont.)

- **Procedure (Performing Job Management Client Functions)**
  - **Verify that a box job (e.g., a box job with status to be modified) is listed in the Currently Selected Job field of the Job Activity Console (Ops Console)**
  - **Click on the Client Tool button**
  - **Enter the number corresponding to the desired function at the "enter an option" prompt**
  - **Enter responses to Job Management Client prompts**

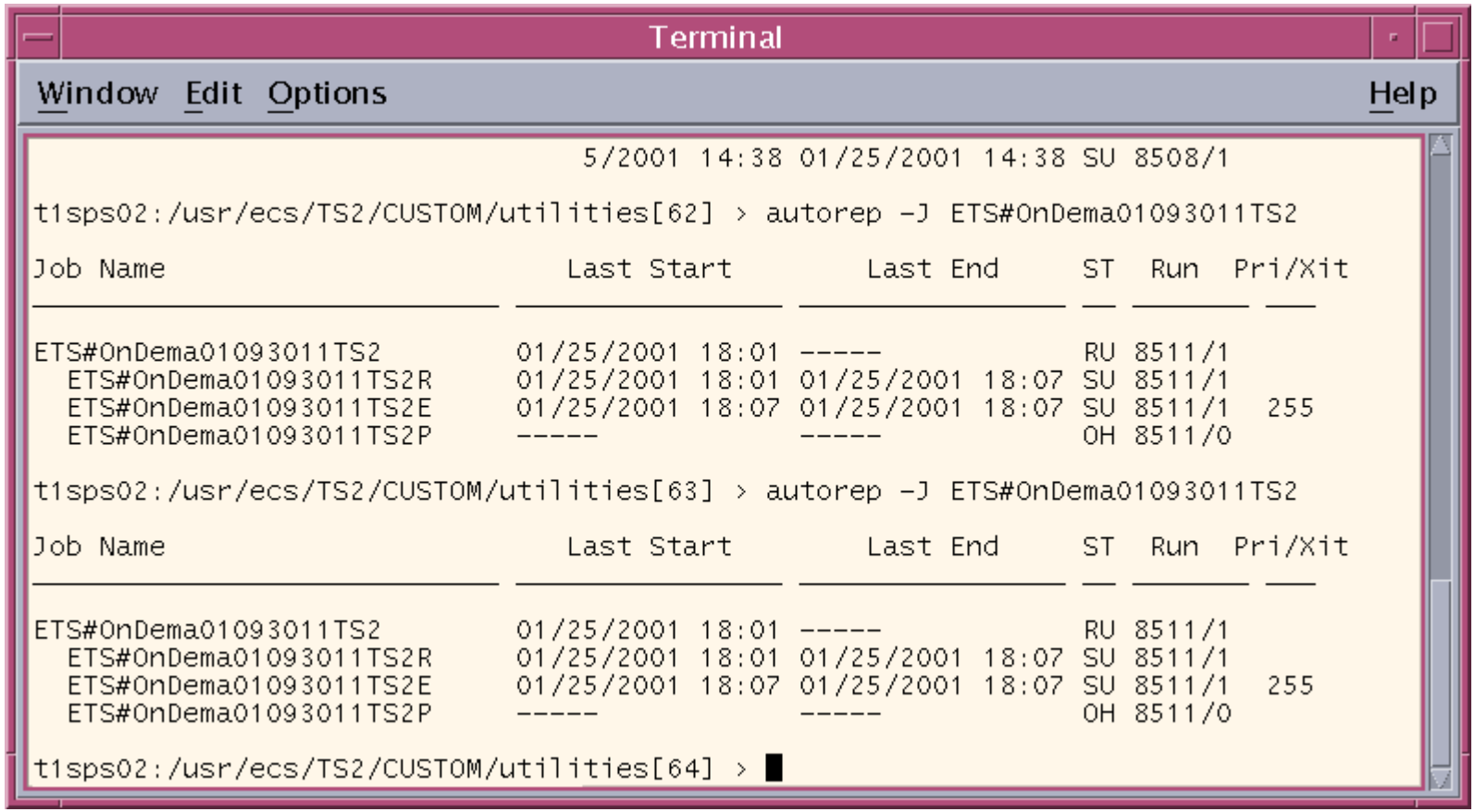
Job Activation User Interface Window



Reviewing Activity and Job Dependency Reports

- **Reviewing a Job Activity Report**
  - **Production Monitor reviews a job activity report to determine...**
    - » **which jobs are currently in the AutoSys queue**
    - » **which jobs have been completed**
    - » **the completion status of jobs that have been completed**
    - » **which jobs are currently running**

Sample Job Activity Report



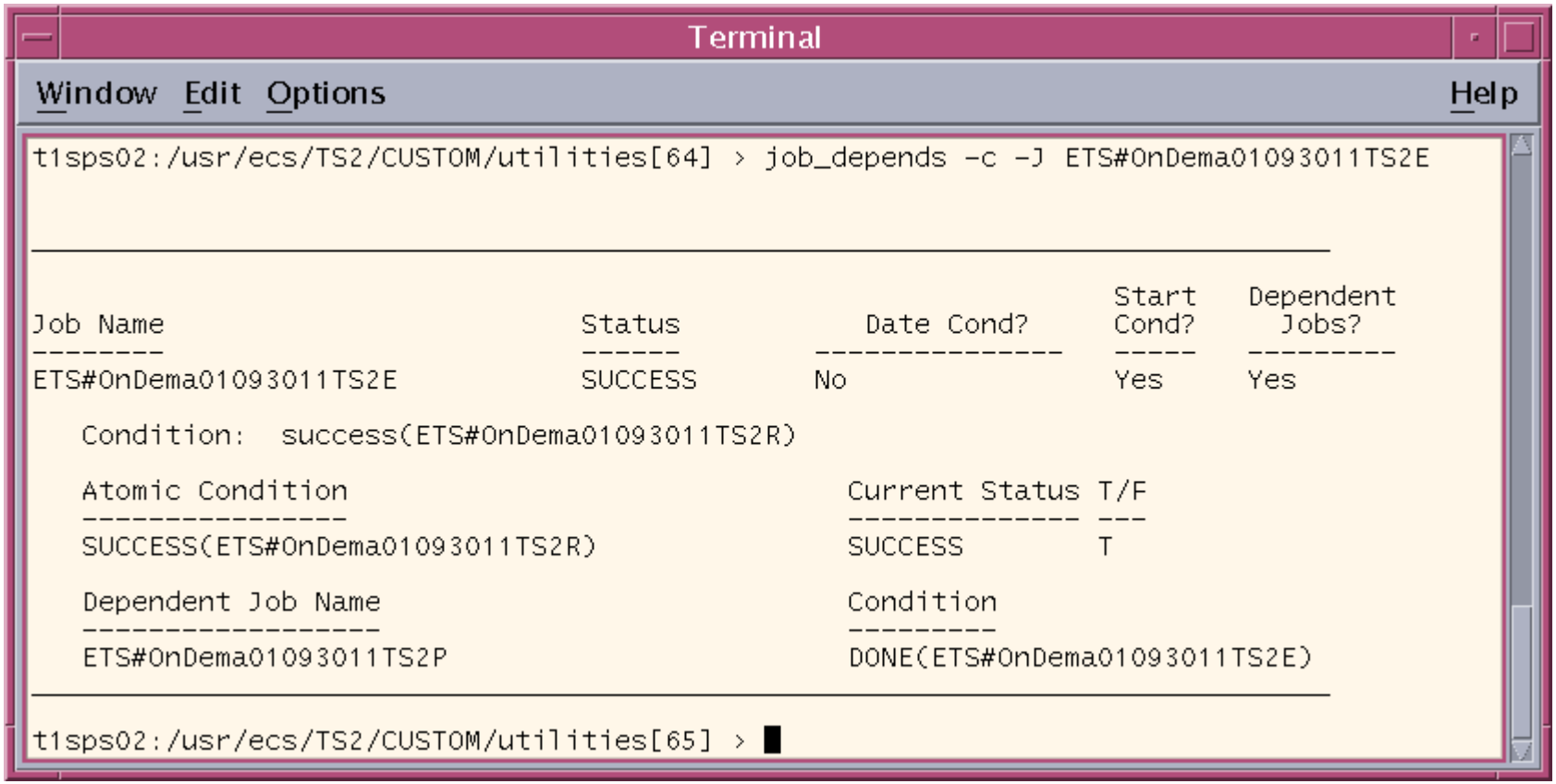
Reviewing Reports (Cont.): Job Activity Report

- **Procedure**
  - Set up AutoSys
  - Type autorep -J ALL unless the command needs to be modified to:
    - » specify a particular job
    - » obtain a machine report
    - » obtain a summary report
    - » obtain a detailed report
    - » obtain a query report
    - » print the document
    - » save the document in a file
  - Review the Job Activity Report to identify job states

Reviewing Reports (Cont.)

- **Reviewing a Job Dependency Report**
  - **Production Monitor reviews a job dependency report using the AutoSys job_depends command**
  - **job_depends command reports information about the dependencies and conditions of jobs**
    - » **current state of a job**
    - » **job's dependencies**
    - » **dependencies and nested hierarchies (for boxes) as specified in the job definition**
    - » **forecast of what jobs will run during a given period of time**

Sample Job Dependency Report



Reviewing Reports (Cont.): Job Dependency Report

- **Procedure**
  - **Set up AutoSys**
  - **Type job_depends -c -J ALL unless the command needs to be modified to:**
    - » **specify a particular job**
    - » **obtain the current condition status**
    - » **obtain the dependencies only**
    - » **obtain the time dependencies**
    - » **print the document**
    - » **save the document in a file**
  - **Review the Job Dependency Repot to determine job states**

Defining and Running Monitors/Browsers

- **Defining Monitors/Browsers**
  - **ECS does not currently support AutoSys monitor/browser capabilities**
    - » **However, they are functional and available**
  - **Production Monitor can use the AutoSys Monitor/Browser GUI to define monitors and browsers**
    - » **With no expectation of ECS support if problems are encountered**
  - **monitor function can limit monitoring to alarms and changes of job status (e.g., from "running" to "success" or "failure")**
  - **browser function can be used to determine the eventual status of jobs run during the preceding shift or day; e.g., which jobs were successful, which jobs failed, and which jobs are still running**

Sample Browser Screen

BROWSER: Browser

```
Code=0
Job: PM1DefAtt#063002000PSP STARTING    02/01/2001 16:14:29    Run# 8582:1
Job: PM1DefAtt#063002000PSP RUNNING     02/01/2001 16:14:33    Run# 8582:1
Job: PM1DefAtt#063002000PSP SUCCESS     02/01/2001 16:15:00    Run# 8582:1  Exit
Code=0
Job: PM1DefAtt#063002000PSE STARTING    02/01/2001 16:15:07    Run# 8582:1
Job: PM1DefAtt#063002000PSE RUNNING     02/01/2001 16:15:09    Run# 8582:1
Job: PM1DefAtt#063002000PSE SUCCESS     02/01/2001 16:15:12    Run# 8582:1  Exit
Code=200
Job: PM1DefAtt#063002000PSp STARTING    02/01/2001 16:15:18    Run# 8582:1
Job: PM1DefAtt#063002000PSp RUNNING     02/01/2001 16:15:21    Run# 8582:1
CHK_MAX_ALARM  Job: PM1DefAtt#063004000PSE Machine: t1spg01
  02/01/2001 16:15:24    Run# 8581:1
Job: PM1DefAtt#063002000PSE FAILURE     02/01/2001 16:15:30    Run# 8582:0  Exit
Code=0
Alarm: JOBFAILURE    Job: PM1DefAtt#063002000PSE 02/01/2001 16:15:34    Run# 858
2:0  Exit Code=0
Job: PM1DefAtt#063002000PSp SUCCESS     02/01/2001 16:15:44    Run# 8582:1  Exit
Code=1
CHK_MAX_ALARM  Job: PM1DefAtt#063002000PSE Machine: t1spg01
  02/01/2001 16:46:09    Run# 8582:1
```

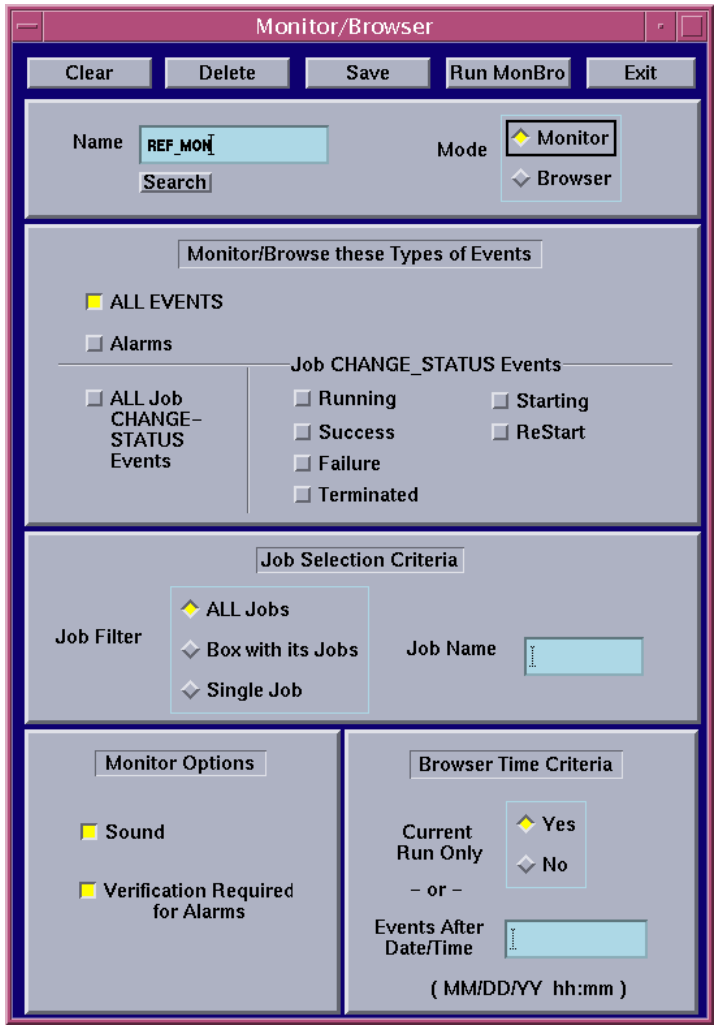
Defining and Running Monitors/Browsers

- **Defining Monitors/Browsers (Cont.)**
  - **When all events for all jobs should be monitored, do not run a monitor**
  - **Instead, display the Event Processor log in real time (using the command autosyslog -e)**
  - **Running a monitor adds another connection to the database and establishes an additional process that is continually polling the database**
    - » **That has a significant impact on system performance**

Monitors/Browsers (Cont.): Defining Monitors/Browsers

- **Procedure**
  - Click on the Monitor/Browser button on the AutoSys GUI Control Panel
  - Type a name for the monitor or browser in the Name field near the top of the GUI
  - Select Types of Events
  - Select Job Status Events
  - Select the desired Job Selection Criteria
  - Select the desired Monitor Options if applicable
  - Select the desired Current Run Time and/or Events After Date/ Time, which are the Browser Time Criteria if applicable
  - Select the desired Mode
  - Save the monitor/browser

Monitor/Browser GUI



Monitors/Browsers (Cont.)

- **Running Monitors/Browsers**
  - **may be run from the Monitor/Browser GUI as described in the preceding procedure**
  - **may be run using a UNIX command**

Monitors/Browsers (Cont.): Running Monitors/Browsers

- **Procedure (Running Monitors/Browsers from the Monitor/Browser GUI)**
  - **Click on the Monitor/Browser button on the AutoSys GUI Control Panel**
  - **If the desired monitor or browser has not been previously defined, define the monitor or browser**
  - **Enter the name of the monitor/browser in the Name field**
  - **Click on the Run MonBro button**
  - **Click on the Exit button to exit from the Monitor/Browser GUI**
  - **Review the monitor/browser results**
  - **Type Ctrl-C in the browser/monitor window to exit from the browser or monitor**

Monitors/Browsers (Cont.): Running Monitors/Browsers

- **Procedure (Running Monitors/Browsers Using UNIX Commands)**
  - **If the desired monitor or browser has not been previously defined, define the monitor or browser as described in the procedure for Defining Monitors/Browsers**
  - **Set up AutoSys**
  - **Type monbro -N *name* &**
  - **Review the monitor/browser results**
  - **Type Ctrl-C in the browser/monitor window to exit from the browser or monitor**

Tuning System Parameters

- System parameters may be subject to control by Configuration Management (CM)
  - When making or requesting a change to system parameters, the CM process at the particular site must be followed (if applicable)
- Two databases where parameters can be set:
  - Configuration Registry database
  - PDPS database

Tuning System Parameters (Cont.)

- **Parameters in the Configuration Registry Database**
  - **Configuration Registry Server provides a single interface (via a Sybase server) for retrieving configuration attribute-value pairs for ECS servers from the Configuration Registry database**
    - » **When ECS servers are started they access the Configuration Registry database to obtain needed configuration parameters**
  - **Database Administrator has access to a Configuration Registry GUI for viewing and editing configuration data in the database**
  - **It is necessary to coordinate with the Database Administrator when changes to configuration parameters are needed**
  - **Changes to configuration-controlled parameters are subject to approval through the site CM process**

Tuning System Configuration Parameters (Cont.)

- **Parameters in the Configuration Registry Database (Cont.)**
  - **Default and adjusted values assigned to system parameters vary from site to site**
  - **For guidance concerning the assignment of values to parameters included in the Configuration Registry refer to document 910-TDA-022, Custom Code Configuration Parameters for ECS**
    - » **Document is available at http://cmdm.east.hitc.com/baseline/ under "Technical Documents"**

Tuning System Parameters (Cont.)

- **Parameters in the Configuration Registry Database (Cont.): Tuning Parameters (parameters whose values may be modified to enhance system functioning or performance)**
  - **AppLogSize [parameter applies to all servers]**
    - » **Maximum size of the application log (ALOG) file for a particular application**
    - » **Recommended size varies considerably depending the nature of the application for which the file is being written**
  - **AppLogLevel [parameter applies to all servers]**
    - » **Level of detail provided in the ALOG file for a particular application**
    - » **Acceptable values are 0, 1, 2, or 3**
    - » **A setting of "0" provides the most data**

Tuning System Parameters (Cont.)

- **Parameters in the Configuration Registry Database (Cont.): Tuning Parameters (Cont.)**
  - **DebugLevel [parameter applies to all servers]**
    - » **Level of detail provided in the debug log file for a particular application**
    - » **Normally acceptable values are 0, 1, 2, or 3**
    - » **A setting of "0" turns off logging; a setting of "3" provides a significant amount of data**
  - **DpPr_MAX_RETRIES [EcDpPrEM and EcDpPrDeletion parameter]**
    - » **Number of retries (e.g., 30) to the Science Data Server for acquires/inserts before giving up**
  - **DpPr_WAIT_PERIOD [EcDpPrEM and EcDpPrDeletion parameter]**
    - » **Time in seconds (e.g., 120) to wait between retries to the Science Data Server**

Tuning System Parameters (Cont.)

- **Parameters in the Configuration Registry Database (Cont.): Tuning Parameters (Cont.)**
  - **DpPrRM_MAX_RETRIES [EcDpPrEM, EcDpPrGE, EcDpPrJobMgmt, EcDpPrDeletion parameter]**
    - » **Maximum number (e.g., 100) of attempts to allocate a computer resource**
  - **DpPrRM_RETRY_PERIOD [EcDpPrEM, EcDpPrGE, EcDpPrJobMgmt, EcDpPrDeletion parameter]**
    - » **Number of seconds (e.g., 120) between retries when trying to allocate a resource**
  - **DpPrMaxConcurrentDPRs [EcDpPrJobMgmt parameter].**
    - » **Maximum allowed jobs**
    - » **Three integer values (e.g., 100 100 100) are assigned to DpPrMaxConcurrentDPRs; the first for routine processing; the second for on-demand processing; and the third for reprocessing jobs**

Tuning System Parameters (Cont.)

- **Parameters in the Configuration Registry Database (Cont.): Tuning Parameters (Cont.)**
  - **DpPrMinConcurrentDPRs [EcDpPrJobMgmt parameter]**
    - » **Minimum allowed jobs**
    - » **NOT CURRENTLY USED**
  - **DpPrAutoSysMaxDPRs [EcDpPrJobMgmt parameter]**
    - » **Maximum number of completed DPRs (i.e., in SUCCESS or FAILEDPGE state) in AutoSys**
    - » **When AutoSys has the maximum number of completed DPRs, the next DPR that succeeds or fails causes the oldest completed DPR to be deleted from AutoSys**
    - » **If the value assigned to DpPrAutoSysMaxDPRs is too low, completed jobs are swept out of AutoSys very quickly, which may not allow the operator enough time to see that the job was completed**

Tuning System Parameters (Cont.)

- **Parameters in the Configuration Registry Database (Cont.): Tuning Parameters (Cont.)**
  - **DpPrDeleteFailedPGEJobs [EcDpPrJobMgmt parameter]**
    - » **If TRUE, failed PGE Jobs are removed by Job Management, as necessary, when space is needed for another job that is ready to run**
    - » **If FALSE (the usual value), failed PGE Jobs are left in AutoSys**
  - **DBConnections [EcPoConnections (includes EcPlSubMgr, EcPlOdMgr, EcDpPrDeletion, EcDpPrJobMgmt and EcDpPrJobMgmtClient) parameter]**
    - » **Number of connections needed by a particular application (e.g., 10 for EcPlOdMgr)**
    - » **Optional parameter that specifies the number of connections to maintain in the connection pool**

Tuning System Parameters (Cont.)

- **Parameters in the Configuration Registry Database (Cont.): Tuning Parameters (Cont.)**
  - **SleepDelayForFailures [EcPlSubMgr parameter]**
    - » **Amount of time in seconds (e.g., 60) to wait before reprocessing failed notifications**
    - » **Sleep delay used by the failed notification thread**
    - » **Less frequent checking can increase speed for the other threads**
  - **SleepDelayForTimers [EcPlSubMgr parameter]**
    - » **Amount of time in seconds (e.g., 60) the Subscription Manager should sleep between checking for expired timers**
    - » **Should be set to the minimum amount of time a timer will be set for at the particular DAAC (min 60 sec)**
    - » **Sleep delay used by the timer checking thread**
    - » **Less frequent checking can increase speed for the other threads**

Tuning System Parameters (Cont.)

- **Parameters in the Configuration Registry Database (Cont.): Tuning Parameters (Cont.)**
  - **SleepDelayForExp [EcPlOdMgr parameter]**
    - » **Sleep delay for expiration thread in seconds (e.g., 86400)**
    - » **Should be considerably greater than the sleep delay for completion threads (SleepDelayForCmp)**
  - **SleepDelayForCmp [EcPlOdMgr parameter]**
    - » **Sleep delay for completion threads in seconds (e.g., 300)**
    - » **Should be considerably less than the sleep delay for expiration threads (SleepDelayForExp)**

Tuning System Parameters (Cont.)

- **Parameters in the Configuration Registry Database (Cont.): Tuning Parameters (Cont.)**
  - **SocketLimit [EcDpPrDeletion, EcDpPrJobMgmt, EcPlOdMgr, EcPlSubMgr parameter]**
    - » **Number of connections (e.g., 200) to a server through the Hubble Space Telescope (HST) sockets middleware**
    - » **Too low a number misses connections**
    - » **Too high a number may adversely affect the memory of the server's host**

Modifying System Parameters (Cont.)

- **Parameters in the Configuration Registry Database (Cont.)**
  - **When the value assigned to a parameter has been changed and saved in the Configuration Registry, the modified value does not take effect until the affected server has been restarted**
  - **Example**
    - » **Debug level for the Subscription Manager log has been changed from "2" to "3" in the Configuration Registry**
    - » **Modification does not affect the recording of data in the log until after a warm restart of the Subscription Manager (at which time the server would read the parameters in the Configuration Registry)**

Modifying System Parameters (Cont.)

- **Parameters in the PDPS Database**
  - **The following two tables in the PDPS database have significant effects on the running of DPRs:**
    - **DpPrPgeLimits - controls where DPRs run**
    - **DpPrClassSchedulingLimits - controls how many DPRs run at a time**
  - **DpPrPgeLimits imposes restrictions on the number of DPRs of a particular PGE that can run simultaneously on the same virtual computer**
  - **A database record defines each pgeId/computerName (PGE/virtual computer) combination that will be run and how many jobs (DPRs) associated with the particular combination can run at the same time**

Modifying System Parameters (Cont.)

- **Parameters in the PDPS Database (Cont.)**
  - **Unless a particular host is specified when a Production Request is created, all jobs in a chain are scheduled to run on the virtual computer(s) specified for the PGE in the DpPrPgeLimits table**
  - **However, if no machine is specified in either the Production Request or in the DpPrPgeLimits table, the jobs run on the computer entered in the PlResourceRequirement table at PGE registration**
  - **An easy way to balance the load on two or more virtual computers is to specify an equal number of pgelds to run on each virtual computer**
    - » **If the number is large (e.g., 10,000), potentially all ready-to-run DPRs specifying the PGE can run and the number is balanced on the valid computers**
    - » **If the number is small (e.g., two per machine), the number of DPRs using the PGE can be throttled, with the excess DPRs being queued**

Modifying System Parameters (Cont.)

- Parameters in the PDPS Database (Cont.)
  - If controlling the total number of DPRs that can run at any one time is considered necessary, the DpPrClassSchedulingLimits table is involved
    - » The table controls the total number of concurrent DPRs scheduled for Routine, Reprocessing and On-demand processing
    - » When a slot is free, all ready-to-run DPRs that have empty slots in DpPrPgeLimits are considered and the DPR with the oldest time stamp in the PlDataProcessingRequest table is selected
  - The DpPrClassSchedulingLimits table has three records, one for each type of processing
    - » If the DpPrClassSchedulingLimits table has no record for a particular type of processing, DPRs of that type are not allowed into AutoSys

Modifying System Parameters (Cont.)

- **Parameters in the PDPS Database (Cont.)**
  - **Values for the maxDprs and minDprs columns in the DpPrClassSchedulingLimits table are loaded at Job Management Server startup using data from the following two configuration parameters:**
    - » **DpPrMaxConcurrentDPRs - maximum allowed jobs**
    - » **DpPrMinConcurrentDPRs - minimum allowed jobs**
  - **Each parameter has three integer values; the first for routine processing; the second for on-demand processing; and the third for reprocessing jobs**
    - » **For example, the Configuration Registry may have the following entries:**

      DpPrMaxConcurrentDPRs = 100 60 40
      DpPrMinConcurrentDPRs = 0 0 0
    - » **Maximum allowed jobs is 100 for routine processing, 60 for on-demand processing, and 40 for reprocessing**
    - » **Minimum allowed jobs is 0 for each type of processing**

Modifying System Parameters (Cont.)

- **Modifying the DpPrPgeLimits and DpPrClassSchedulingLimits Tables (PDPS Database)**
  - **Either the DpPrPgeLimits table or the DpPrClassSchedulingLimits table can be loaded by running the EcDpPrLoadTable.pl script from the Job Management Client tool (using the appropriate option)**
    - » **The Job Management Client tool is accessed through the AutoSys Job Activity Console**
    - » **The EcDpPrLoadTable.pl script loads values from an input data file**
    - » **Instructions for using the script are available in the EcDpPrLoadTable.README file in the /usr/ecs/MODE/CUSTOM/data/DPS directory on the Queuing Server host**
    - » **The same directory has a template for constructing the necessary input data file**

Modifying System Parameters (Cont.)

- **Modifying the DpPrPgeLimits and DpPrClassSchedulingLimits Tables (Cont.)**
  - **The Job Management Client tool has the following options for modifying the DpPrPgeLimits table or the DpPrClassSchedulingLimits table:**

    8) Change Max Concurrent Jobs for PGE Limits table

    - » **For DpPrPgeLimits table modifications**

    9) Change Max/Min Dprs for Job Class

    - » **For DpPrClassSchedulingLimits table modifications.**
  - **For detailed instructions on modifying the tables using the Job Management Client tool refer to the procedure for Performing Job Management Client Functions**
  - **An alternative method of modifying the tables is to create one's own load script using SQL statements**

Modifying System Parameters (Cont.)

- **Modifying the DpPrPgeLimits and DpPrClassSchedulingLimits Tables (Cont.)**
  - **It is acceptable to add pgeId entries for a machine, add new machines to the DpPrPgeLimits table, or change the maximum number of DPRs that can concurrently execute in DpPrClassSchedulingLimits**
  - **However, values for the number of currently scheduled or running DPRs in the tables must not be changed**
  - **Also, note that the DpPrPgeLimits table can be empty but DpPrClassSchedulingLimits must be fully populated**
    - » **As previously mentioned, default values for the maxDprs and minDprs columns in the DpPrClassSchedulingLimits table are loaded at Job Management Server startup using data from configuration parameters in the Registry database**

Tuning System Parameters (Cont.)

- **Monitoring the Load on Processing Resources**
  - **Production Planner and Production Monitor should work with the Resource Planner to make optimum use of processing resources**
    - » **Resource Planner allocates the disk partitions, CPUs, and RAM available for processing among the active modes (e.g., OPS, TS1, TS2)**
    - » **Production Planner and Production Monitor monitor the load on the processing resources**

Tuning System Parameters (Cont.)

- **Monitoring the Load on Processing Resources (Cont.)**
  - **Resource Planner assigns the bulk (typically 60% - 80%) of the processing resources to the OPS mode**
    - » **The remainder of the processing assets are divided among the modes used for SSI&T and new version software checkout**
  - **The Production Planner and Production Monitor monitor the load on the processing resources to identify whether the actual load is appropriately distributed among modes**
    - » **They inform the Resource Planner of under- or over-use of resources as allocated**

Tuning System Parameters (Cont.)

- **Monitoring the Load on Processing Resources (Cont.)**
  - **Disk space allocation**
    - » **Disk space allocated to OPS mode is likely to be used to capacity**
    - » **Disk space assigned to the other two modes may not fill up**
  - **CPU allocation**
    - » **There is no one-to-one mapping of CPU allocation with actual CPUs on the science processor**
    - » **The operating system(OS) takes care of true CPU and RAM allocation**
    - » **Actual CPU usage during processing is limited by OS**
    - » **If ten CPUs have been specified for a mode, only ten DPRs can be running the Execute job at a given time**
    - » **What is really being defined is the maximum number of DPRs that will execute at a given time**

Tuning System Parameters (Cont.)

- **Monitoring the Load on Processing Resources (Cont.)**
  - **CPU allocation (Cont.)**
    - » **CPUs can be over-allocated or under-allocated as necessary to get the most out of the CPUs on each science processor**
    - » **If monitoring indicates that the processor is underused when OPS mode is at full processing capacity, the number of CPUs allocated to OPS mode could probably be increased**
    - » **If the science processor is at full capacity when OPS mode is at full processing capacity (and the processor may be overworked) the number of CPUs allocated to OPS mode should be reduced**
  - **Random-access memory (RAM) allocation**
    - » **Subject to the same considerations as CPUs**
    - » **RAM can be over-allocated or under-allocated to get the most out of the memory on each sci. processor**

Tuning System Configuration Parameters (Cont.)

- **Strategies for Tuning**
  - **Section includes...**
    - » **Scenario that demonstrates how DPRs might be processed under a particular set of conditions**
    - » **Some strategies for tuning the system**
  - **The processing conditions include the following types of items:**
    - » **The total number of jobs allowed into AutoSys**
    - » **The number of CPUs available for processing**
    - » **Characteristics of the PGEs to be processed**

Tuning System Configuration Parameters (Cont.)

- **Scenario (DPR Processing)**
  - **The total number of jobs (DPRs) allowed into AutoSys is controlled by the DpPrPgeLimits table in the PDPS database**
  - **An example of some of the types of data maintained in the DpPrPgeLimits table is shown in the following table:**

computerName [Virtual Computer]	pgeId	maxConcurrent [DPRs]
A	1	20
B	1	20
A	2	20
B	2	20

Tuning System Configuration Parameters (Cont.)

- **Scenario (DPR Processing) (Cont.)**
  - **Scenario assumes that each of the virtual computers (i.e., A and B) listed in the preceding table has 16 CPUs**
    - » **32 CPUs total**
  - **Relevant PGE characteristics are shown in the table that follows:**

PGE	# CPUs Used	Average Execution Time	Average Stage Time	Destage Time
1	1	5 minutes	5 minutes	5 minutes
2	1	60 minutes	5 minutes	5 minutes

  - **Scenario assumes that 100 DPRs of each type (i.e., PGE 1 and PGE 2 - 200 DPRs total) are ready to run and are released at once into AutoSys**

Tuning System Configuration Parameters (Cont.)

- Scenario (DPR Processing) (Cont.)
  - Eighty (80) DPRs enter AutoSys
  - The remaining 120 DPRs are queued, with their assignments already made:
    - » Machine (Virtual Computer) A
      - 20 PGE 1s start staging
      - 30 PGE 1s are queued on Machine A
      - 20 PGE 2s start staging
      - 30 PGE 2s are queued on Machine A
    - » Machine (Virtual Computer) B
      - 20 PGE 1s start staging
      - 30 PGE 1s are queued on Machine B
      - 20 PGE 2s start staging
      - 30 PGE 2s are queued on Machine B

Tuning System Configuration Parameters (Cont.)

- Scenario (DPR Processing) (Cont.)
  - After about five (5) minutes, all 80 DPRs that were staging have finished staging and are ready for execution
    - » However, only 32 CPUs are available
  - The first 32 DPRs that ask for CPUs get them and start running
    - » Sixteen (16) on Machine A and sixteen (16) on Machine B
  - Forty-eight (48) DPRs are waiting
    - » Assuming that parameters in the Registry database are set as follows:
      - DpPrRM_RETRY_PERIOD = 120 seconds
      - DpPrRM_MAX_RETRIES = 100

      the waiting DPRs keep trying every two minutes for up to 100 times each before timing out (after 200 min.)
    - » In this example timing out is a real possibility

Tuning System Configuration Parameters (Cont.)

- **Scenario (DPR Processing) (Cont.)**
  - **The quick jobs complete processing after five (5) minutes, freeing up sixteen (16) CPUs**
    - » **In the current example, the sixteen (16) CPUs are subsequently occupied with about eight (8) five-minute PGEs and eight (8) 60-minute PGEs because CPUs are given randomly to whichever DPR gets back first to asking for them after waiting for the retry period (i.e., 120 seconds)**
    - » **Priorities are not used**
    - » **At first, there was a 50:50 ratio of fast:slow DPRs, now there is a 25:75 ratio of fast:slow**
    - » **After another five (5) minutes, the ratio becomes 12.5:87.5 fast:slow, so 87.5 % of the CPUs are occupied by 60-minute DPRs**

Tuning System Configuration Parameters (Cont.)

- **Scenario (DPR Processing) (Cont.)**
  - **The 60-minute DPRs tend to dominate the CPUs**
    - » **After one (1) hour the first batch of sixteen (16) 60-minute PGEs vacates the CPUs to be replaced by eight (8) five-minute PGEs and eight (8) 60-minute PGEs, but the five-minute PGEs become extinguished again by the slow ones**
    - » **If the staging and destaging times were not the same (so the DPRs didn't have the same opportunity to hit the execution stage at the same time) the scenario would proceed differently**

Tuning System Configuration Parameters (Cont.)

- **Strategies for Tuning the System:**
  - **Limit the number of DPRs through the use of the DpPrPgeLimitsTable**
  - **Increase the declared number of CPUs for the processors to more than the actual number (overallocate CPUs)**
  - **Create new virtual computers (assigning CPUs on the processors to them) and assign (via the DpPrPgeLimits table) PGEs to run on the new virtual computers**

Tuning System Configuration Parameters (Cont.)

- **Strategies for Tuning the System (Cont.):**
  - **Limit the number of DPRs through the use of the DpPrPgeLimitsTable**
    - » **In the example if the number of slow DPRs allowed into AutoSys is less than the number of CPUs, there is always a channel for the fast jobs to squeeze through**
    - » **The big disadvantage to this approach is that the slow jobs are also being prevented from staging**
  - **Increase the declared number of CPUs for the processors to more than the actual number (overallocate CPUs)**
    - » **This approach allows more of each type of PGE into the science processors**
    - » **The disadvantage is that it could overwhelm the science computers; however, they are kept busy**

Tuning System Configuration Parameters (Cont.)

- **Strategies for Tuning the System (Cont.):**
  - Create new virtual computers (assigning CPUs on the processors to them) and assign (via the DpPrPgeLimits table) PGEs to run on the new virtual computers
    - » This approach is another way to guarantee bandwidth (CPUs) to PGEs
    - » The disadvantage of this approach is that some CPUs could remain idle, not being seen by one of the virtual computers
    - » In the past, there may have also been some code problems with supporting this, but those difficulties should have been resolved

Tuning System Configuration Parameters (Cont.)

- **Strategies for Tuning (Cont.)**
  - **Probably some combination of the first two strategies is best**
    - » **Increase the number of declared CPUs to be more than the total number of slow jobs allowed into AutoSys, always leaving some CPUs for a channel of fast jobs**
    - » **The total number of faster-moving jobs should be increased to make sure that there is always be a queue of them available to get their channel occupied**
  - **The staging and destaging times have to be accounted for and this could change things in terms of using the DpPrPgeLimits table and the number of CPUs per processor to tune the job flow**
  - **It is important to perform regular garbage collection on all of the virtual computers**

Tuning System Configuration Parameters (Cont.)

- **AutoSys Database Maintenance Time**
  - Once a day the Event Processor (also known as the AutoSys daemon) goes into an internal database maintenance cycle
  - During this time, it does not process any events
  - It waits for the maintenance activities to be completed before resuming normal operations
  - Time of day for start-up of the maintenance cycle is pre-set to 3:30 AM
  - AutoSys database maintenance cycle takes approximately one minute
  - If it is necessary to change the time when the maintenance cycle occurs, whoever has "write" access to the configuration file can reset it
    - » Preferably to a time when there is minimal activity

Tuning System Configuration Parameters (Cont.)

- **Procedure (Changing the AutoSys Database Maintenance Time)**
  - **Access a terminal window logged in to the Queuing Server host**
  - **Change directory to the subdirectory (e.g., autouser) containing the config.*AutoSysInstance* file**
  - **Use the vi editor to find DBMaintTime=03:30 and replace the existing time with the desired time in 24 hour format (hh:mm)**
  - **Save the edited file**

Troubleshooting Processing Problems

- **Troubleshooting:**

  process of identifying the source of problems on the basis of observed trouble symptoms

Troubleshooting Processing Problems

- **Problems with production planning can usually be traced to…**
  - **some part of the Data Processing Subsystem**
  - **problems in other ECS subsystems, including (but not necessarily limited to):**
    - » **Planning Subsystem (PLS)**
    - » **Data Server Subsystem (DSS)**
    - » **Communications Subsystem (CSS)**
- **Fault Recovery**
  - **Discussed in the section on Troubleshooting Production Planning Problems**

Troubleshooting Processing Problems (Cont.)

- **Troubleshooting table**
  - **describes actions to be taken in response to some common Processing problems**
  - **if the problem cannot be identified and fixed without help within a reasonable period of time, call the help desk and submit a trouble ticket in accordance with site Problem Management policy**

Troubleshooting Processing Problems (Cont.)

Symptom	Response
Unable to log in to the Queuing Server host (e.g., g0sps06).	Check with the Operations Controller/System Administrator to ensure that the host is “up."
GUI not displayed when the start-up script has been properly invoked.	Ensure that the DISPLAY variable was set properly. [For detailed instructions refer to the procedure for **Launching the AutoSys GUI Control Panel** (previous section of this lesson).]
Entire processing system hangs (no jobs change state over time).	1. Ensure that it is possible to connect to the necessary hosts and servers (listed in Table 6). [For detailed instructions refer to the section on **Checking Connections to Hosts/Servers** (subsequent section of this lesson).] 2. If hosts/servers are all “up,” perform the procedure for **Responding to Hanging of the Processing System** (subsequent section of this lesson).
Jobs are activated but do not get started in AutoSys.	Refer to the procedure for **Responding to Failure of Jobs to Start in AutoSys** (subsequent section of this lesson).
AutoSys box job hangs (does not change state over time).	Refer to the procedure for **Handling a Box Job that is Hanging in AutoSys** (subsequent section of this lesson).
“Preprocess” function fails (job either does not change state over time or has turned red on JobScape or TimeScape).	Refer to the procedure for **Handling a Hanging or Failed Preprocessing Job** (subsequent section of this lesson).

Troubleshooting Processing Problems (Cont.)

Symptom	Response
"Execute" job hangs (job has turned orange or oscillates between orange and green on JobScape or TimeScape).	Refer to the procedure for **Handling a Hanging Execution Job** (subsequent section of this lesson).
"Execute" job fails (job has turned red on JobScape or TimeScape).	Refer to the procedure for **Handling a Failed Execution Job** (subsequent section of this lesson).
"Postprocess" job fails (job has turned red on JobScape or TimeScape).	Refer to the procedure for **Handling a Failed Postprocessing Job** (subsequent section of this lesson).
Both the "Execute" and "Postprocess" jobs fail (jobs have both turned red on JobScape or TimeScape).	Refer to the procedure for **Handling Failure of Both Execution and Postprocessing Jobs** (subsequent section of this lesson).
On-Demand Processing Request fails.	Refer to the procedure for **Handling a Failed On-Demand Processing Request** (subsequent section of this lesson).
Other problems.	Check the log files (e.g., EcDpPrJobMgmt.ALOG, EcDpPrJobMgmt.Debug.log, EcDpPrDeletion.ALOG, *DPR*#.ALOG, *DPR*#.err, etc.) in the /usr/ecs/*MODE*/CUSTOM/logs directory for error messages. [For detailed instructions refer to the procedure for **Checking Log Files** (subsequent section of this lesson).]

Troubleshooting Processing Problems (Cont.)

- **Production Processing Troubleshooting Procedures**
  - Checking AutoSys Status
  - Checking the AutoSys Log
  - Checking Job Management Server Status
  - Checking to Determine Whether the DPR Is Waiting in the AutoSys Queue
  - Checking to Determine Whether AutoSys Is Full
  - Responding to a Condition Where a DPR Was Released But Failed Due to a JIL Failure
  - Handling Subscription Server Problems
  - Responding to a DPR That Was Released But Failed Due to an AutoSys ID Failure

Troubleshooting Processing Problems (Cont.)

- **Production Processing Troubleshooting Procedures (Cont.)**
  - Responding to a DPR That Was Released But Failed Due to Invalid DPR
  - Responding to a DPR That Was Released But Failed to Be Received by Job Management Server
  - Handling a Hanging Allocation Function
  - Running Execution Management Outside of AutoSys
  - Handling a Failed Allocation Function
  - Force-Starting a Job
  - Responding to a Restart of a Job That Fails Although All Known Problems Have Been Corrected
  - Handling a Failed Staging Function

Troubleshooting Processing Problems (Cont.)

- **Production Processing Troubleshooting Procedures (Cont.)**
  - Cleaning Up the DPS File Tables
  - Handling a Failed Preprocessing Job
  - Responding to Execution Job That Has Failed and the DPR Has Gone into "Failed-PGE" Processing
  - Handling a Failed Postprocessing Job
  - Handling Failure of Both Execution and Postprocessing Jobs
  - Handling a Failed Insertion Function
  - Handling a Failed Deallocate Function
  - Responding to a DPR that Failed in OdMgr because the PGE ID Could Not Be Found
  - Checking Log Files

Troubleshooting Processing Problems (Cont.)

- **Responding to hanging of the processing system**
  - **Checking AutoSys status**
  - **Checking the AutoSys log**

Troubleshooting Processing Problems (Cont.)

- **Responding to failure of jobs to start in AutoSys**
  - **Checking Job Management Server status or checking to determine whether the DPR is waiting in the AutoSys queue (never got released)**
    - » **Checking Job Management Server Status**
    - » **Checking to Determine Whether the DPR Is Waiting in the AutoSys Queue**
    - » **Using ISQL to Check Database Tables**
    - » **Checking to Determine Whether AutoSys Is Full**
    - » **Responding to a Condition Where a DPR Was Released But Failed Due to a JIL Failure**
    - » **Handling Subscription Server Problems**

Troubleshooting Processing Problems (Cont.)

- **Responding to failure of jobs to start in AutoSys (Cont.)**
  - **Responding to a DPR that was released but failed due to an AutoSys ID failure**
  - **Responding to a DPR that was released but failed due to invalid DPR**
  - **Responding to a DPR that was released but failed to be received by Job Management Server**

Troubleshooting Processing Problems (Cont.)

- **Responding to a single DPS job that has failed or is hanging**
  - **Handling a Box Job that is Hanging in AutoSys**
  - **Handling a Hanging Allocation Function**
  - **Running Execution Management Outside of AutoSys**
  - **Handling a Failed Allocation Function**
  - **Force-Starting a Job**
  - **Responding to a Restart of a Job That Fails Although All Known Problems Have Been Corrected**
  - **Handling a Hanging Staging Function**
    - » **Perform the Handling a Hanging Allocation Function procedure**

Troubleshooting Processing Problems (Cont.)

- **Responding to a single DPS job that has failed or is hanging (Cont.)**
  - **Handling a Failed Staging Function**
  - **Cleaning Up the DPS File Tables**
  - **Handling a Failed Preprocessing Job**
  - **Handling a Hanging Execution Job**
    - » **Perform the Checking AutoSys Status procedure**
  - **Handling a Failed Execution Job**
    - » **Perform the Checking AutoSys Status procedure**
  - **Responding to Execution Job and/or Postprocessing Job That Have (Has) Failed**
  - **Responding to Execution Job That Has Failed and the DPR Has Gone into "Failed-PGE" Processing**
  - **Handling a Failed Postprocessing Job**

Troubleshooting Processing Problems (Cont.)

- **Responding to a single DPS job that has failed or is hanging (Cont.)**
  - **Handling Failure of Both Execution and Postprocessing Jobs**
  - **Handling a Failed Insertion Function**
  - **Handling a Failed Deallocate Function**

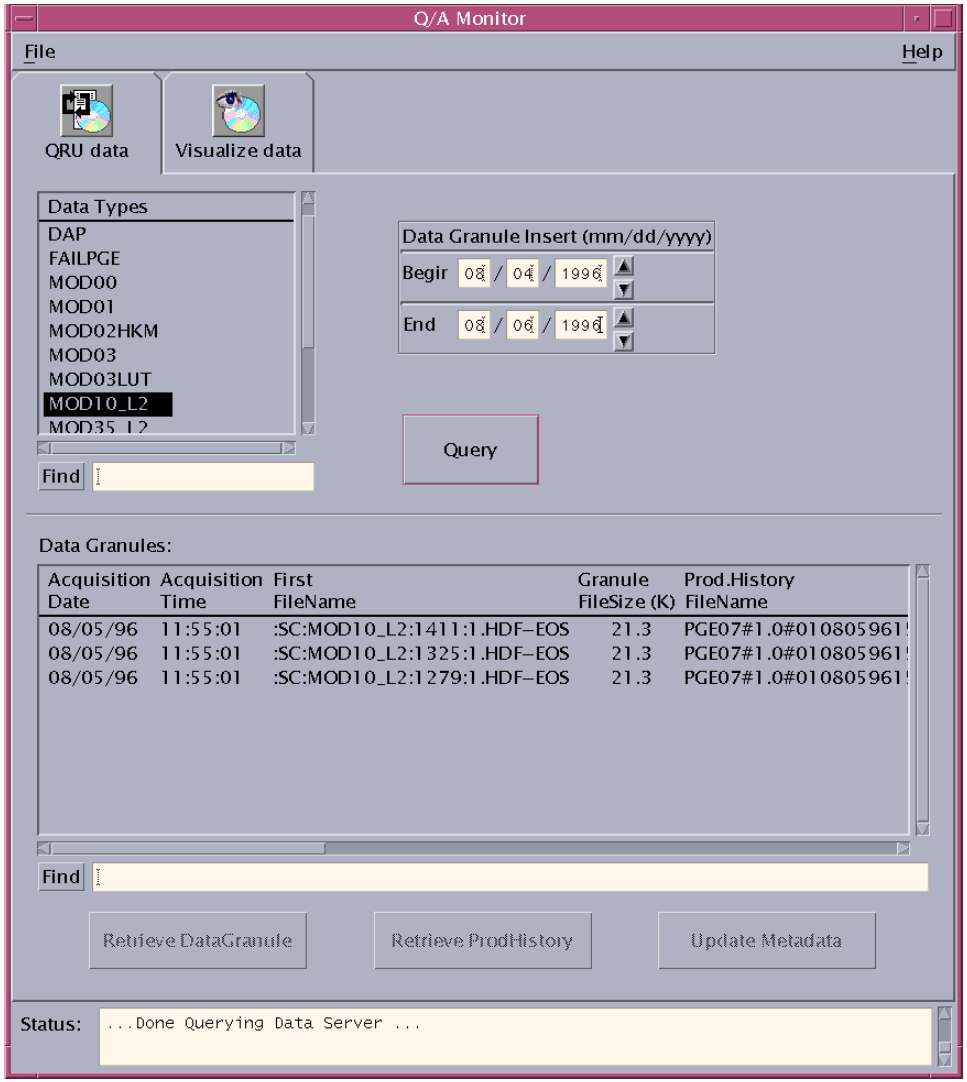
Troubleshooting Processing Problems (Cont.)

- **Handling a failed On-Demand Processing Request**
  - **Responding to a DPR that failed in OdMgr because the PGE ID could not be found**
- **Checking log files**

Launching the QA Monitor

- **Procedure**
  - Access a terminal window logged in to the Planning/Management Workstation
  - Set the ECS_HOME environmental variable if necessary
  - Change directory to the subdirectory (e.g., utilities) containing the QA Monitor start script
  - Start the QA Monitor GUI in the appropriate mode

QA Monitor GUI: QRU Data Tab



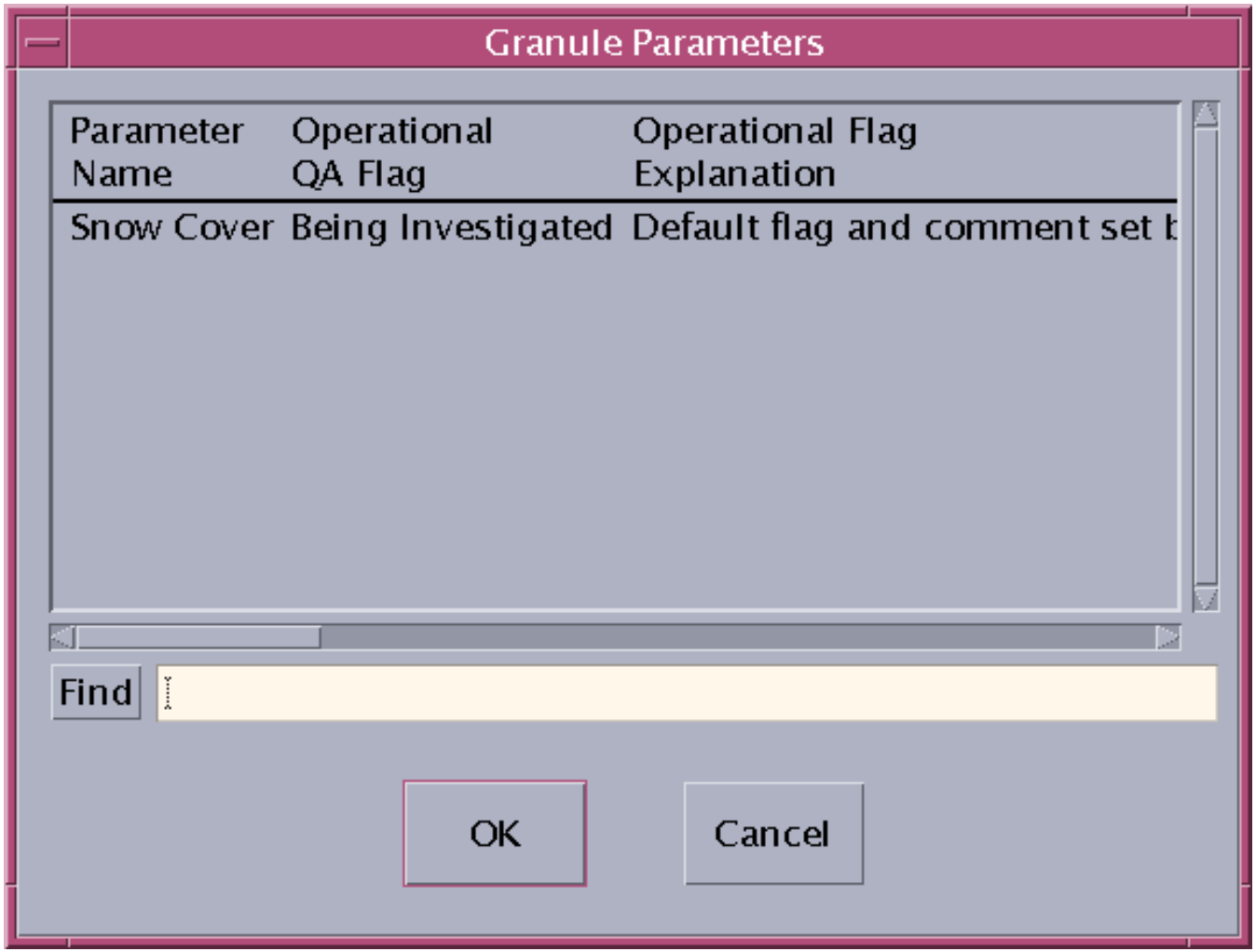
Performing Science Product Quality Assurance (QA)

- Uses the QA Monitor application
- Science Computing Facility Personnel
  - responsible for performing QA of their products
- Production Monitor
  - updates QA metadata in response to a request from SCF personnel to set the metadata flags on specified granule(s)

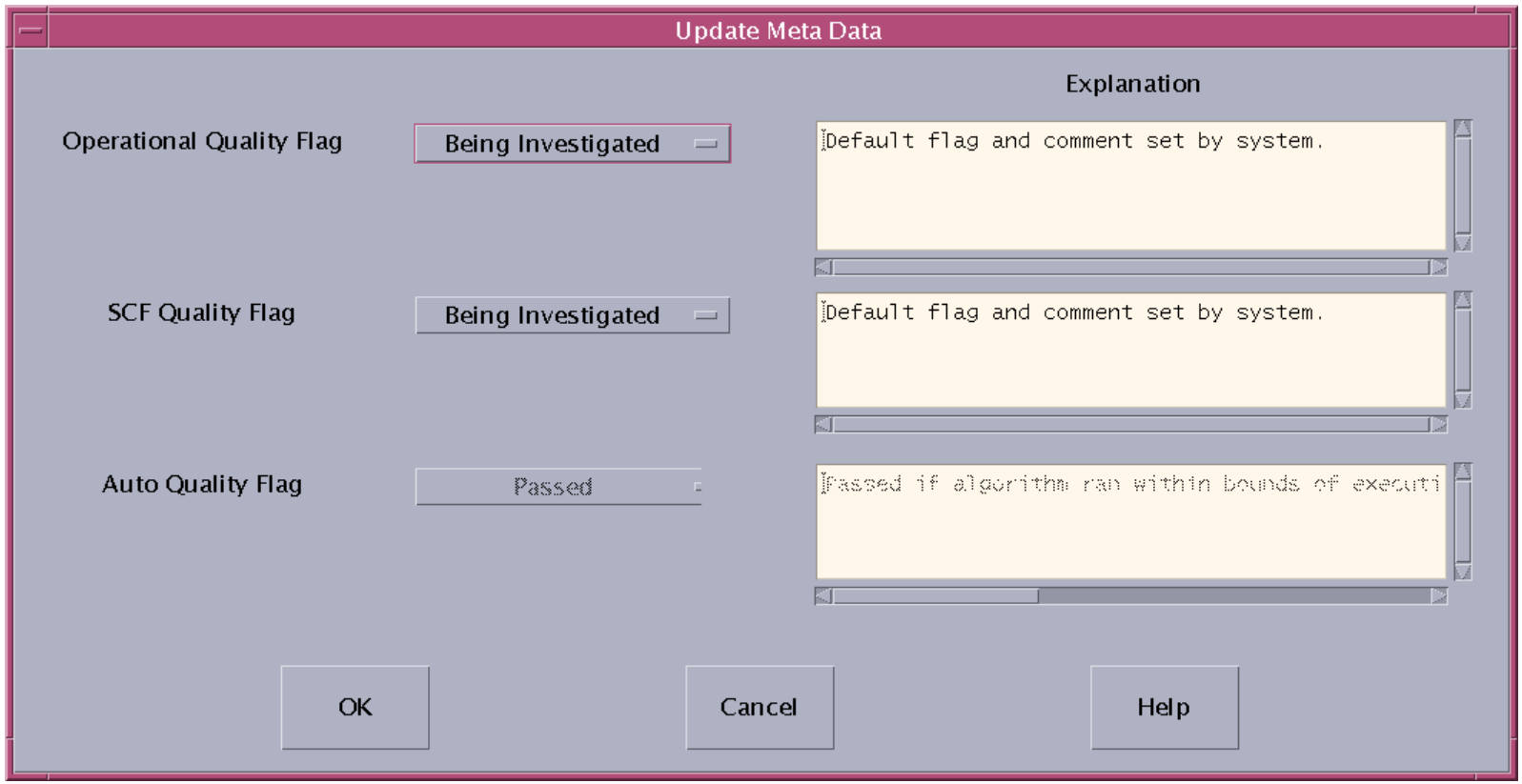
Updating Quality Assurance (QA) Metadata

- **Procedure**
  - **Set up and query the database using the QA Monitor GUI**
  - **Select the granule with QA metadata to be updated**
  - **Set the operational and SCF quality flags to the appropriate value (as specified by the SCF personnel)**
  - **Verify that the flags have actually been set in the database by repeating the set-up and query processes**

QA Monitor GUI: Granule Parameters Window



QA Monitor GUI: Update Meta Data Window



Regenerating Granules

- **Produce replacements for previously generated granules that have been lost or corrupted due to failure in the ECS archive**
- **General Process:**
  - **Retrieve the Production History file (PH) for the lost granule to determine parameters for the generation of replacement granules**
  - **Create Production Requests for the generation of replacement granules**
  - **Create and activate a Production Plan that includes the Production Requests for the generation of replacement granules**
  - **Prepare (if applicable) a "PDPS Residual Granules List," which identifies granules that either cannot or should not be regenerated at the DAAC**
  - **Some granules do need not be reproduced; e.g., if there is a more recent version of the product**

Regenerating Granules (Cont.)

- **Procedure**
  - Retrieve the Production History tar file for each granule in the Granules for PDPS Re-Generation list that needs to be reproduced
  - Launch the SSIT Manager GUI
  - Re-register the PGE (if not currently registered)
  - Launch the Production Request Editor
  - Create a Production Request for the relevant PGE/version/profile ID
  - Launch the Planning Workbench
  - Create and activate a production plan that includes the newly created Production Request(s)
  - Send the PDPS Residual Granules list to the originator of the Granules for PDPS Re-Generation list